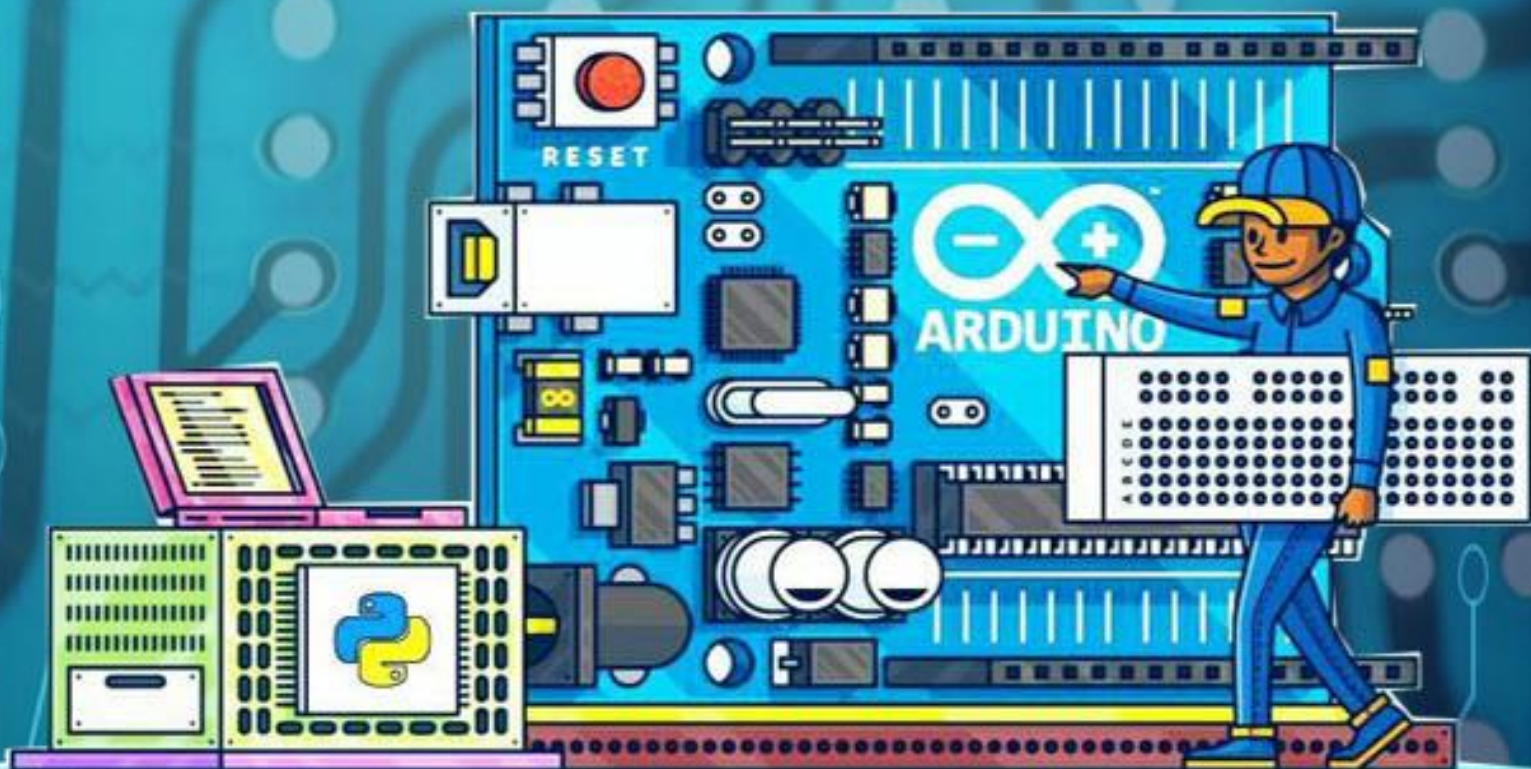


Programming Arduino IDE



គាំទ្រផ្នែកការលើកកម្ពស់ការប្រើប្រាស់បច្ចេកវិទ្យា ប្រព័ន្ធគ្រប់គ្រង និងសុវត្ថិភាព



Programming Arduino IDE

គាំទ្រវិកាលើការរៀបរៀងដោយ មូលនិធិស្រាវជ្រាវ គំនិតច្នៃប្រឌិត និងនវានុវត្តន៍
រៀបរៀងដោយលោក **គាន ម៉ីនណា**

គណៈកម្មការរៀបរៀង និងត្រួតពិនិត្យ

១. លោក គឹម សុផា	ប្រធានដេប៉ាតឺម៉ង់អេឡិចត្រូនិច	ប្រធាន
២. លោក ឈីវ ឫទ្ធី	គ្រូបង្រៀនឯកទេសអេឡិចត្រូនិច	អនុប្រធាន
៣. លោក ខ្លឹម លាស់	គ្រូបង្រៀនឯកទេសអេឡិចត្រូនិច	សមាជិក
៤. អ.ស ប៉ាង ណារីន	អនុប្រធានការិយាល័យ (ឯកទេសភាសាខ្មែរ)	សមាជិក

មុព្វកថា

ដំណើរអភិវឌ្ឍន៍នៃព្រះរាជាណាចក្រកម្ពុជានៅក្នុងយុគសម័យទំនើបនេះ ជាមេរៀនដ៏ជោគជ័យ បំផុតមួយដែលចាប់បួសគល់ចេញពីការបញ្ចប់របបប្រល័យពូជសាសន៍ ការបញ្ចប់សង្គ្រាម ការផ្សះផ្សារជាតិ ការកសាងមូលដ្ឋានវិស័យសេដ្ឋកិច្ចនិងស្ថេរភាព និងការអភិវឌ្ឍសេដ្ឋកិច្ច។ នៅក្រោយពេលដែលសេដ្ឋកិច្ចត្រូវបានកើតឡើងដោយបរិបូណ៌នៅឆ្នាំ១៩៩៨ កម្ពុជាទទួលបានកំណើនសេដ្ឋកិច្ចខ្ពស់ គឺប្រមាណ ៨% ក្នុងមួយឆ្នាំ។ លើសពីនេះទៀត អត្រានៃភាពក្រីក្រត្រូវបានកាត់បន្ថយពីប្រមាណ៥៣% នៅឆ្នាំ ២០០៤ មកនៅទាបជាង១០% នៅឆ្នាំ២០១៩។ ដំណើរនៃការអភិវឌ្ឍជាតិជាសកម្មភាពដែលបន្តទៅមុខជាប់ជានិច្ច ហើយគោលនយោបាយថ្មីៗដែលមានលក្ខណៈអន្តរវិស័យគ្របដណ្តប់ ក៏កំពុងលេចរូបរាងឡើង ដើម្បីតម្រង់ទិសកម្ពុជាឆ្ពោះទៅកាន់ប្រទេសមានប្រាក់ចំណូលមធ្យមកម្រិតខ្ពស់នៅឆ្នាំ ២០៣០ និងឈានឡើងជាប្រទេសមានប្រាក់ចំណូលខ្ពស់ នៅឆ្នាំ២០៥០។ ការប្រែប្រួលឆាប់រហ័សនៃនិម្មាបនកម្មពិភពលោកនិងតំបន់ រួមទាំងទំនាក់ទំនងភូមិសាស្ត្រនយោបាយ បានផ្តល់កាលានុវត្តភាពសម្រាប់ការអភិវឌ្ឍឧស្សាហកម្មនៅកម្ពុជា ដែលត្រូវបានរាជរដ្ឋាភិបាលចាត់ទុកជាមូលដ្ឋានគ្រឹះនៃកំណើនសេដ្ឋកិច្ចកម្ពុជា។ រាជរដ្ឋាភិបាលកម្ពុជាបាន និងកំពុងបន្តពង្រឹង និងអភិវឌ្ឍវិស័យអប់រំឆ្ពោះទៅរកការស្រាវជ្រាវ និងនវានុវត្តន៍ ដើម្បីពង្រឹងសមត្ថភាពនិងជំនាញរបស់ធនធានមនុស្សនៅកម្ពុជា ឱ្យស្របទៅនឹងបរិបទថ្មីនៃការអភិវឌ្ឍ ជាពិសេសការពង្រឹងសហគ្រិនភាពក្នុងការរៀបចំម៉ូដែលធុរកិច្ចថ្មីៗ។ ដើម្បីចាប់យកកាលានុវត្តភាពពីបដិវត្តន៍ឧស្សាហកម្មទី៤ និងសេដ្ឋកិច្ចឌីជីថលដែលកំពុងផុសផុលឡើង ប្រព័ន្ធអេកូឡូស៊ីដែលបង្កលក្ខណៈអំណោយផលដល់ការបង្កើតថ្មី នវានុវត្តន៍ ការស្រាវជ្រាវ និងអភិវឌ្ឍន៍ត្រូវតែមានការកែលម្អ។

បណ្តាប្រទេសនៅទ្វីបអាស៊ីកំពុងនាំមុខក្នុងការវិនិយោគលើការស្រាវជ្រាវនិងអភិវឌ្ឍ ដោយមានភាគហ៊ុនប្រមាណ៤៤% នៃការវិនិយោគទាំងមូលរបស់ពិភពលោក។ ប្រទេសចិនកំពុងបន្តកសាងហេដ្ឋារចនាសម្ព័ន្ធនៃការវិនិយោគលើការស្រាវជ្រាវនិងអភិវឌ្ឍ ក៏ដូចជាសមត្ថភាពមនុស្ស។ ផ្ទុយទៅវិញ ប្រទេសនៅទ្វីបអាមេរិកខាងត្បូង និងអាហ្វ្រិក កំពុងស្ថិតនៅឆ្ងាយពីការវិនិយោគនេះ ហើយជាលទ្ធផល ប្រទេសទាំងនោះក៏ពុំមានកំណើនសេដ្ឋកិច្ចគួរឱ្យកត់សម្គាល់ដែរ។ ទុនវិនិយោគសរុបលើការស្រាវជ្រាវនិងអភិវឌ្ឍរបស់ប្រទេសនៅទ្វីបអាមេរិកខាងត្បូងនិងអាហ្វ្រិក មានប្រមាណ៥%នៃការវិនិយោគទាំងមូលរបស់ពិភពលោក ក្នុងពេលដែលតំបន់ទាំង២នេះមានប្រជាជនប្រមាណ២០%នៃប្រជាជនពិភពលោក។ ប្រទេសចំនួន៦ដែលមានលំដាប់ខ្ពស់ជាងគេនៅក្នុងការវិនិយោគលើការស្រាវជ្រាវនិងអភិវឌ្ឍ រួមមានសហរដ្ឋអាមេរិក ចិន ជប៉ុន អាល្លឺម៉ង់ ឥណ្ឌា និងកូរ៉េខាងត្បូង ដែលស្មើនឹងប្រមាណ៧០%នៃទុនវិនិយោគសរុបរបស់ពិភពលោក។

តើចំណេះដឹង ផលិតផល និងសេវាកម្មថ្មីទាំងនេះកើតឡើងពីអ្វី? ហើយកើតឡើងដោយរបៀបណា? ព្រះរាជាណាចក្រកម្ពុជាកំពុងតែកសាងមូលដ្ឋានសម្រាប់ការត្រៀមខ្លួនទទួល និងប្រកួតប្រជែងក្នុងយុគសម័យបដិវត្តន៍ឧស្សាហកម្មទី៤ នៅក្នុងសេដ្ឋកិច្ចដែលផ្អែកលើពុទ្ធិ ហើយដែលប្រការនេះ

ចាំបាច់តម្រូវឱ្យពលរដ្ឋកម្ពុជា ត្រូវក្លាយខ្លួនជាពលរដ្ឋឌីជីថល ពលរដ្ឋសកល និងពលរដ្ឋដែលប្រកបដោយការទទួលខុសត្រូវ ដែលមានសមត្ថភាពក្នុងការផលិត ចែកចាយ និងប្រើប្រាស់ពុទ្ធដើម្បីទទួលបានផល និងរួមចំណែកក្នុងកំណើន។ ធនាគារពិភពលោកបានធ្វើការកត់សម្គាល់តាំងពីឆ្នាំ ២០០២នូវបម្លាស់ប្តូរនៃមូលដ្ឋានសេដ្ឋកិច្ច ពីសេដ្ឋកិច្ចដែលពឹងផ្អែកលើកម្លាំងពលកម្ម និងធនធានអតិកម្ម (Labour and Resource Based Economy) ទៅកាន់សេដ្ឋកិច្ចដែលពឹងផ្អែកលើពុទ្ធិ (Knowledge Based-Economy) ដែលក្នុងន័យនេះ ពុទ្ធិគឺជាគន្លឹះនៃការអភិវឌ្ឍ។ អាស្រ័យហេតុនេះនៅលើគន្លងដែលកម្ពុជាកំពុងធ្វើដំណើរឆ្ពោះទៅកាន់សេដ្ឋកិច្ចឌីជីថល សង្គមកម្ពុជាត្រូវតែមានសមត្ថភាពក្នុងការផលិត ជ្រើសរើស បន្សុំ បង្កើតមុខរបរ និងប្រើប្រាស់ពុទ្ធិ ដើម្បីរក្សានិរន្តរភាពនៃកំណើន និងកែលម្អជីវភាពរស់នៅ។ សមត្ថភាពទាំងនេះ អាចកើតឡើងនៅពេលពលរដ្ឋកម្ពុជាមានឱកាសក្នុងការទទួលបានបទពិសោធន៍ពីការស្រាវជ្រាវ ការបណ្តុះគំនិតច្នៃប្រឌិត និងការស្វែងរកនវានុវត្តន៍។

កំណែទម្រង់វិស័យអប់រំ គឺជាការត្រួសត្រាយមាត់សម្រាប់ដំណើរឆ្ពោះទៅកាន់សង្គមប្រកបដោយពុទ្ធិ និងប្រជាពលរដ្ឋប្រកបដោយភាពរស់រវើក។ តាមរយៈមូលដ្ឋានអប់រំ សង្គមប្រកបដោយពុទ្ធិនឹងប្រមូលផ្តុំ បង្កើត និងចែករំលែក ទៅកាន់សមាជិកក្នុងសង្គមនូវសម្បទាអប់រំ ពិសេសគឺពុទ្ធិសម្បទាក្នុងបុព្វហេតុនៃមនុស្សជាតិនិងឧត្តមប្រយោជន៍នៃប្រទេស។ សង្គមប្រកបដោយពុទ្ធិ គឺពុំគ្រាន់តែជាសង្គមដែលសម្បូរព័ត៌មានប៉ុណ្ណោះទេ តែជាសង្គមដែលប្រជាពលរដ្ឋអាចធ្វើបរិវត្តកម្មព័ត៌មានទៅជាមូលធនប្រកបដោយប្រសិទ្ធភាព។ ការរីកចម្រើនទៅមុខជាលំដាប់នៃបច្ចេកវិទ្យានិងតំណភ្ជាប់ បានពង្រីកព្រំដែននៃការចូលទៅកាន់ និងការទទួលបានព័ត៌មានជាសកល ហើយដែលក្នុងន័យនេះ ការអប់រំនឹងបន្តវិវត្តទៅមុខនិងមានការផ្លាស់ប្តូរ។ សង្គមមួយដែលមានអំណាន និងរបាំបុរេលក្ខខណ្ឌនៃជីវភាពប្រចាំថ្ងៃនៃប្រជាពលរដ្ឋ ពេលនោះបំណិននៃអំណាន និពន្ធ និងការគណនាលេខនព្វន្ត គឺជាចលករនៃការរៀនរបស់សិស្ស។ ធាតុដ៏ចម្បងមួយដែលស្ថិតនៅក្នុងការកសាងសង្គមដែលប្រកបដោយពុទ្ធិគឺសៀវភៅសិក្សា ហើយការរៀបរៀង និពន្ធ និងកែលម្អសៀវភៅសិក្សាជាប្រចាំ គឺជានវានុវត្តន៍នៃវិស័យអប់រំដែលនាំទៅរកការសិក្សាពេញមួយជីវិត ការអភិវឌ្ឍសម្បទាអប់រំ និងការចែករំលែកចំណេះដឹង។ មូលដ្ឋានអប់រំ ជាពិសេសគឺគ្រឹះស្ថានឧត្តមសិក្សាត្រូវមានតួនាទីដែលប្រកបដោយការឆ្លើយតប ចំពោះតម្រូវការខាងលើនេះ។ សាស្ត្រាចារ្យ អ្នកស្រាវជ្រាវ និងបុគ្គលិកអប់រំត្រូវបន្តសិក្សាជាប់ជានិច្ច តាមរយៈការរៀបរៀង និពន្ធ និងកែលម្អសៀវភៅសិក្សា ហើយដែលសៀវភៅសិក្សាទាំងនេះនឹងក្លាយជាស្ថាននៃទំនាក់ទំនងរវាងនវានុវត្តន៍នៃបច្ចេកវិទ្យា និងការរៀននិងបង្រៀននៅក្នុងថ្នាក់រៀន។

សង្គមដែលប្រកបពុទ្ធិ ក៏ជាសង្គមដែលបណ្តុះឱ្យមានរចនាសម្ព័ន្ធទំនន់នៃសេដ្ឋកិច្ចដែលពឹងផ្អែកលើពុទ្ធិដែរ។ ឧទាហរណ៍ជាក់ស្តែងនៃបែបផែននេះរួមមាន Silicon Valley នៃសហរដ្ឋអាមេរិក សួនឧស្សាហកម្មវិទ្យាសាស្ត្រអាកាសយានយន្តនិងយានយន្តនៅទីក្រុង Munich ប្រទេសអាល្លឺម៉ង់ តំបន់ជីវបច្ចេកវិទ្យានៅក្រុង Hyderabad ប្រទេសឥណ្ឌា តំបន់ផលិតគ្រឿងអេឡិចត្រូនិកនិងសារគមនាគមន៍ឌីជីថលនៅទីក្រុង Seoul ប្រទេសកូរ៉េខាងត្បូង ក៏ដូចជាសួនឧស្សាហកម្មថាមពល និងឥន្ធនគីមីសាស្ត្រនៃប្រទេសប្រេស៊ីល ហើយក៏នៅមានទីក្រុងនៃប្រទេសជាច្រើនទៀតនៅលើពិភពលោក។ លក្ខណៈសម្បត្តិ

នៃទីក្រុងទាំងនេះគឺការប្រើប្រាស់និន្នាការនៃការអភិវឌ្ឍដែលជំរុញ និងតម្រង់ទិសដោយចំណេះដឹង ហើយដែលចំណេះដឹងទាំងនោះកើតចេញជាដំបូងពីការវិនិយោគទៅលើគ្រឹះស្ថានឧត្តមសិក្សា ស្ថាប័ន ស្រាវជ្រាវ មជ្ឈមណ្ឌលឧត្តមភាពនៃជំនាញជាន់ខ្ពស់ ការប្រកួតប្រជែងដោយគុណាធិបតេយ្យ និងជាពិសេស គឺការបណ្តុះវប្បធម៌អំណាននិងនិពន្ធសៀវភៅ។ ល្បឿននៃការរីកចម្រើនផ្នែកពុទ្ធិ និងបច្ចេកវិទ្យាកំពុងមាន សន្ទុះលឿនជាងអ្វីដែលសិស្ស និងនិស្សិតអាចទទួលបានពីគ្រូនៅគ្រឹះស្ថានសិក្សា ដែលធ្វើឱ្យគោលដៅ នៃការអប់រំនៅពេលបច្ចុប្បន្ននេះ មានការប្រឈមខ្លាំងជាងពេលណាទាំងអស់។ ឧទាហរណ៍ ក្នុងមួយឆ្នាំ មានសៀវភៅជាង២,២លានចំណងជើង ត្រូវបានសរសេរនិងបោះពុម្ព ដែលក្នុងនោះប្រទេសចិនមាន ៤៤០ពាន់ ចំណែកឯសហរដ្ឋអាមេរិកមាន៣០៥ពាន់ និងប្រទេសរុស្ស៊ីមាន១២០ពាន់ចំណងជើង។

ខណៈពេលដែលបច្ចេកវិទ្យាកំពុងរីកចម្រើនជារៀងរាល់ថ្ងៃ មធ្យោបាយសម្រាប់អំណានក៏មាន ច្រើនជាងម្រឹមសម្រាប់សិស្ស-និស្សិត និងសាធារណៈជន រួមមានការអានសៀវភៅ ការអានលើឧបករណ៍ អេឡិចត្រូនិក ការអានដោយប្រើទូរសព្ទវីឡា និងការអានលើកុំព្យូទ័រ ដែលសុទ្ធសឹងជាមធ្យោបាយ សំខាន់ៗដែលនាំអ្នកអានទាំងឡាយឱ្យសម្រេចគោលបំណងអានរបស់ខ្លួន។ ម្យ៉ាងវិញទៀត អំណាន ដោយប្រើមធ្យោបាយបច្ចេកវិទ្យាទំនើប ចំណាយពេលតិច ងាយស្រួលអាន និងជួយដល់បរិស្ថាន មួយកម្រិតទៀត។ នាពេលបច្ចុប្បន្ន សិស្ស-និស្សិត និងសាធារណៈជនកម្ពុជាដែលស្រឡាញ់អំណាន កំពុងតែប្រើប្រាស់មធ្យោបាយអំណានទាំងនេះ។ បើយើងក្រឡេកមើលទៅប្រទេសជប៉ុនលឿន ទោះបីជា បច្ចេកវិទ្យារីកចម្រើនខ្លាំងយ៉ាងណា អំណានតាមរយៈសៀវភៅនៅតែមានសន្ទុះដដែល។ ម្យ៉ាងវិញទៀត បច្ចេកវិទ្យាអានបែបទំនើបតាមរយៈឧបករណ៍ទំនើប អាស្រ័យលើលទ្ធភាពនៃធនធានអប់រំឌីជីថល និង មាតិកាឌីជីថលគ្រប់គ្រាន់ដែលបានផលិត និងបង្ហោះចែកចាយសម្រាប់អំណាន។

ក្នុងបរិបទកម្ពុជា ជាពិសេសក្នុងបរិការណ៍នៃការផ្ទុះរីករាលដាលនៃជំងឺកូវីដ-១៩ ក្រសួងអប់រំ យុវជន និងកីឡា បានជំរុញឱ្យមានបរិវត្តកម្មឌីជីថលនៅក្នុងអេកូស៊ីស្តែមនៃការអប់រំ ជាពិសេសការអប់រំ តាមប្រព័ន្ធអេឡិចត្រូនិក និងការអប់រំពីចម្ងាយដើម្បីលើកកម្ពស់អំណាន តាមរយៈការផលិតមាតិកា ឌីជីថលដែលមានភាពចម្រុះ ការកសាងសមត្ថភាពផ្នែកគំណាត់និងវេទិកាឌីជីថល ការពង្រីកវិសាលភាព នៃមជ្ឈមណ្ឌលទិន្នន័យ និងការលើកកម្ពស់គុណភាពនៃការផលិតធនធានអប់រំឌីជីថល គួបជាមួយ ការចែកសន្លឹកកិច្ចការឱ្យសិស្សយកទៅរៀននៅផ្ទះ និងការចុះទៅជួបជាមួយសិស្សជាបណ្តុំនៅតាម សហគមន៍។ ក្នុងន័យលើកកម្ពស់អំណាន និងភាពសម្បូរបែបនៃធនធានសៀវភៅសិក្សា ឱ្យកាន់តែ មានប្រសិទ្ធភាពនិងភាពសក្តិសិទ្ធិ និងផ្តល់ឱកាសអំណានកាន់តែច្រើនថែមទៀតដល់សិស្សានុសិស្ស និស្សិត និងសាធារណៈជន ក្រសួងអប់រំ យុវជន និងកីឡាលើកទឹកចិត្តនូវចំណុចមួយចំនួនដូចខាង ក្រោម៖

១. សាស្ត្រាចារ្យ អ្នកស្រាវជ្រាវ និងបុគ្គលិកអប់រំ សូមបន្តនិងបង្កើនការបោះពុម្ពស្នាដៃបន្ថែម ទៀត ដើម្បីធ្វើឱ្យធនធានសម្រាប់អំណានកាន់តែសម្បូរបែប ជាពិសេសធនធានអំណានជា ខេមរភាសា

២. គ្រឹះស្ថានឧត្តមសិក្សា សូមផ្តល់លទ្ធភាពគ្រប់បែបយ៉ាង ដើម្បីឱ្យបុគ្គលិកអប់រំគ្រប់លំដាប់ថ្នាក់ និងនិស្សិតគ្រប់កម្រិតសិក្សាអាចចូលរួមអាន និងសិក្សាស្រាវជ្រាវតាមគ្រប់លទ្ធភាពជាមួយធនធានអំណាន ជាពិសេសការរៀបចំឱ្យមានពេលវេលាសម្រាប់សហសិក្សា និងអំណានក្នុងបណ្ណាល័យ
៣. សាស្ត្រាចារ្យតាមមុខវិជ្ជា និងអ្នកស្រាវជ្រាវតាមជំនាញឬវិស័យ ត្រូវរៀបចំដំណើរការរៀនបង្រៀន និងស្រាវជ្រាវដែលមានដាក់បញ្ចូលកិច្ចការស្វ័យសិក្សា សហសិក្សា ឬការស្រាវជ្រាវបណ្ណាល័យដែលតម្រូវឱ្យនិស្សិត ត្រូវអាននិងស្រាវជ្រាវជាមួយធនធានអំណាន
៤. គ្រឹះស្ថានឧត្តមសិក្សា និងមជ្ឈមណ្ឌលស្រាវជ្រាវ ត្រូវខិតខំឱ្យអស់លទ្ធភាពក្នុងការបង្កើតបណ្ណាល័យ មជ្ឈមណ្ឌលរក្សាឯកសារ ឬមជ្ឈមណ្ឌលអប់រំឌីជីថលជាដើម ដើម្បីឱ្យបុគ្គលិកអប់រំគ្រប់លំដាប់ថ្នាក់និងនិស្សិតគ្រប់កម្រិតសិក្សាអាចទទួលបាន និងស្វែងរកប្រភពសម្រាប់អំណានកាន់តែសម្បូរបែប និងមានភាពបត់បែន ឆ្លើយតបតាមតម្រូវការអ្នកអាន
៥. និស្សិតគ្រប់កម្រិតសិក្សាត្រូវខិតខំនិងចំណាយពេលអាន និងចាត់ទុកវប្បធម៌ និងអកប្បកិរិយាអំណានជាផ្នែកមួយ នៃពេលវេលានិងភាពស៊ីវិល័យនៃជីវិតប្រចាំថ្ងៃ
៦. បងប្អូនជនរួមជាតិ ដែលជាមាតាបិតា ឬអ្នកអាណាព្យាបាល សូមជួយជំរុញនិងបង្កលក្ខណៈកាន់តែច្រើនថែមទៀត ជាពិសេសការលើកចំណាយនៅក្នុងគ្រួសារសម្រាប់ការទិញសម្ភារៈសិក្សា សៀវភៅអាន និងឧបករណ៍សម្រាប់អំណានដល់កូនៗ ដែលចាត់ទុកជាការវិនិយោគមួយដ៏សំខាន់ សម្រាប់ បង្កើនចំណេះដឹង និងអនាគតរបស់ពួកគេ។

ដោយមានការគាំទ្រពីក្រសួងសេដ្ឋកិច្ច និងហិរញ្ញវត្ថុ នៅឆ្នាំ២០២០ ក្រសួងអប់រំ យុវជន និងកីឡា បានបង្កើតមូលនិធិស្រាវជ្រាវ គំនិតច្នៃប្រឌិត និងនវានុវត្តន៍ ដែលហៅកាត់ថា “មូលនិធិ ស.គ.ន.” និងហៅជាភាសាអង់គ្លេសថា The Research Creativity and Innovation Fund ដែលហៅកាត់ជាភាសាអង់គ្លេសថា “RCI Fund”។ គោលដៅចម្បងនៃមូលនិធិនេះ គឺរួមចំណែកលើកកម្ពស់វប្បធម៌នៃការស្រាវជ្រាវ បំផុសគំនិតច្នៃប្រឌិត និងជំរុញការធ្វើនវានុវត្តន៍ ដើម្បីជាប្រយោជន៍ដល់វិស័យអប់រំ យុវជន និងកីឡា ដែលឆ្លើយតបទៅនឹងទិដ្ឋភាពលកម្ម និងសាកលភាវូបនីយកម្ម។ មូលនិធិ ស.គ.ន. បានសម្រេចកំណត់ប្រធានបទ ជាអាទិភាពសម្រាប់ការគាំទ្រដោយមូលនិធិចំនួន៣ រួមមានឌីជីថលនីយកម្មសម្រាប់បដិវត្តឧស្សាហកម្ម៤.០ (Digitalization for IR.4.0) ការស្រាវជ្រាវអនុវត្តលើវិស័យកសិកម្ម (Applied Agricultural Research) និងការស្រាវជ្រាវគរុកោសល្យសតវត្សទី២១ (21st Century Pedagogy Research)។

ដោយមានការធ្វើអាទិភាពរូបនីយកម្មទៅលើទិសដៅ នៃការប្រើប្រាស់ថវិកាមូលនិធិសម្រាប់ឆ្នាំ២០២០ ក្រសួងសេដ្ឋកិច្ច និងហិរញ្ញវត្ថុ និងក្រសួងអប់រំ យុវជន និងកីឡា បានផ្តល់ការគាំទ្រដល់ការរៀបរៀង និងរៀបចំ និងកែលម្អ សៀវភៅសិក្សា (Text book) ដែលនឹងត្រូវប្រើប្រាស់នៅកម្រិតឧត្តមសិក្សា។ គោលបំណងនៃការរៀបរៀង និងកែលម្អ សៀវភៅសិក្សានៅកម្រិតឧត្តមសិក្សា គឺដើម្បីបង្កើនបរិមាណ លើកកម្ពស់គុណភាព និងពង្រីកសមធម៌នៃធនធានសិក្សាជាខេមរភាសា ជូនដល់និស្សិត

ដែលកំពុងបន្តការសិក្សា និងត្រៀមខ្លួនធ្វើការស្រាវជ្រាវនៅកម្រិតឧត្តមសិក្សា។ លើសពីនេះទៀត ការរៀបរៀង និងនីតិវិធី និងកែលម្អសៀវភៅសិក្សានៅកម្រិតឧត្តមសិក្សា មានគោលដៅដូចខាងក្រោម ៖

- ឆ្លើយតបជាបន្ទាន់ចំពោះការខ្វះខាតធនធានសិក្សា ដែលជាតម្រូវការសិក្សារបស់និស្សិត នៅកម្រិតឧត្តមសិក្សា
- លើកកម្ពស់ទំនើបការបន្ថែម និងឧត្តមានុវត្តន៍នៃការរៀននិងបង្រៀន និងការស្រាវជ្រាវ នៅលើមុខវិជ្ជា កម្មវិធីសិក្សា ឬមុខជំនាញជាក់លាក់
- បង្កើនភាពស៊ីជម្រៅក្នុងការកសាងវិជ្ជាជីវៈនិងបទពិសោធន៍សម្រាប់ឋានៈសាស្ត្រាចារ្យ និង អ្នកស្រាវជ្រាវ
- រួមចំណែកដល់ការកសាងភាពជាសហគមន៍វិជ្ជាជីវៈ ការចែករំលែកបទពិសោធន៍ និងវប្បធម៌ នៃការរៀបរៀង និងនីតិវិធី និងកែលម្អសៀវភៅសិក្សានៅកម្រិតឧត្តមសិក្សា។

ក្រសួងអប់រំ យុវជន និងកីឡា បានវាយតម្លៃខ្ពស់ចំពោះការបោះជំហានប្រកបដោយមនសិការ វិជ្ជាជីវៈនៃគ្រឹះស្ថានឧត្តមសិក្សា និងបុគ្គលិកអប់រំទាំងអស់ ក្នុងការរៀបចំ រៀបរៀង និងនីតិវិធី និងកែលម្អ សៀវភៅសិក្សា ដើម្បីបង្កើនបរិមាណ លើកកម្ពស់គុណភាព និងពង្រឹងសមធម៌នៃធនធានសិក្សាជា ខេមរភាសា ជូននិស្សិតដែលកំពុងបន្តការសិក្សា និងត្រៀមខ្លួនធ្វើការស្រាវជ្រាវនៅកម្រិតឧត្តមសិក្សា។ សៀវភៅសិក្សាជាផ្នែកមួយនៃការទទួលស្គាល់គុណភាពអប់រំនៃគ្រឹះស្ថានឧត្តមសិក្សា និងជាធនធាន សិក្សាដែលជាមូលដ្ឋានមួយដ៏សំខាន់ ក្នុងការគាំទ្រដល់ការបង្រៀន និងរៀន ហើយត្រូវមានបរិមាណ គ្រប់គ្រាន់ ឆ្លើយតបទៅនឹងកម្មវិធីអប់រំ និងតម្រូវការសិក្សាស្រាវជ្រាវ។ ជាគោលការណ៍ គ្រឹះស្ថានឧត្តម សិក្សាទាំងអស់ ត្រូវមានសៀវភៅសិក្សាដែលប្រើជាគោលសម្រាប់មុខវិជ្ជានីមួយៗ។ ចំនួនសៀវភៅសិក្សា ដែលគ្រប់គ្រាន់សម្រាប់ការស្រាវជ្រាវ និងការសិក្សារបស់និស្សិត ត្រូវមានយ៉ាងតិចមួយចំណងជើង ក្នុងមួយមុខវិជ្ជា ហើយត្រូវតម្កល់យ៉ាងតិច២ច្បាប់ នៅក្នុងបណ្ណាល័យ ឬអាចរកបានតាមប្រព័ន្ធអេឡិចត្រូនិក។ ក្រសួងអប់រំ យុវជន និងកីឡា លើកទឹកចិត្តបន្ថែមទៀតជូនដល់គ្រឹះស្ថានឧត្តមសិក្សារដ្ឋ និង ឯកជនដែលបានស្នើសុំថវិកាមូលនិធិរួច សូមចូលរួមបន្ថែមទៀតដើម្បីបង្កើនចំនួនចំណងជើងសៀវភៅ។ ចំណែកគ្រឹះស្ថានឧត្តមសិក្សារដ្ឋ និងឯកជនដែលពុំទាន់បានដាក់ពាក្យស្នើសុំសូមចូលរួម ដើម្បីជា គុណប្រយោជន៍ដល់តម្រូវការដ៏ទទួល និងថ្លៃថ្នូរនៃនិស្សិតកម្ពុជាក្នុងការសិក្សា និងស្រាវជ្រាវនៅកម្រិត ឧត្តមសិក្សា។

សេចក្តីបញ្ជាក់
នៃមូលនិធិស្រាវជ្រាវ គំនិតច្នៃប្រឌិត និងនវានុវត្តន៍

សៀវភៅសិក្សានេះជាលទ្ធផលនៃការស្នើសុំអនុវត្តថវិកាមូលនិធិស្រាវជ្រាវ គំនិតច្នៃប្រឌិត និងនវានុវត្តន៍ ក្នុងគម្រោងរៀបរៀង និងរៀបចំ និងកែលម្អសៀវភៅសិក្សា ដែលនឹងត្រូវប្រើប្រាស់នៅកម្រិតឧត្តមសិក្សា។ សៀវភៅសិក្សានេះ ត្រូវបានរៀបរៀង និងរៀបចំ ឬកែលម្អដោយមានការធានាអះអាងថាជាស្នាដៃរបស់អ្នកនិពន្ធជ្រាល់ និងបានឆ្លងកាត់ត្រួតពិនិត្យ ផ្តល់យោបល់ និងវាយតម្លៃដោយក្រុមប្រឹក្សាអប់រំក្រុមប្រឹក្សាស្រាវជ្រាវ ឬក្រុមប្រឹក្សាដែលមានតម្លៃស្មើនៃគ្រឹះស្ថានឧត្តមសិក្សា និងតាមរយៈកិច្ចសន្យាដែលបានធ្វើឡើង និងដែលបានតម្កល់ទុកនៅមូលនិធិស្រាវជ្រាវ គំនិតច្នៃប្រឌិត និងនវានុវត្តន៍។ រាល់ខ្លឹមសារ ការបកស្រាយ និងរូបភាព គឺជាជំហរនិងទស្សនៈផ្ទាល់របស់អ្នកនិពន្ធ ហើយពុំឆ្លុះបញ្ចាំង ឬជាតំណាងដល់មូលនិធិស្រាវជ្រាវ គំនិតច្នៃប្រឌិត និងនវានុវត្តន៍ នៃក្រសួងអប់រំ យុវជន និងកីឡាឡើយ។

អារម្ភកថា

សៀវភៅនេះរៀបរៀងឡើងដើម្បីជាគំនិតមួយក្នុងការរៀនមូលដ្ឋានគ្រឹះ ដែលទាក់ទងជាមួយមុខវិជ្ជា Microcontroller សម្រាប់សិស្សចាប់ពីថ្នាក់បរិញ្ញាបត្ររងឡើងទៅ និងជាជំនួយក្នុងការធ្វើគម្រោង ឬ រៀបចំបញ្ហាផ្សេងៗ។ នៅក្នុងសៀវភៅនេះបាននិយាយពីវិធីប្រើកម្មវិធី Arduino ក្នុងកម្រិតមូលដ្ឋាន ដែលរៀនតាំងពីកម្រិតភាសា C /C++ សម្រាប់សរសេរគ្រប់គ្រង Arduino ។ ក្រៅពីនេះក៏មានរៀនពីវិធីតភ្ជាប់ ការវាស់តម្លៃ Input និង Output ដែលមានការលើកជាឧទាហរណ៍ការប្រើ Sensor ដែលពេញនិយម ការបង្ហាញទិន្នន័យនៅលើ Serial Monitor, LCD, OLED ផងដែរ។ បន្ថែមពីលើនេះក៏មានរៀបរាប់ពី របៀបទាញយកកម្មវិធី Arduino IDE របៀបដំឡើង Arduino IDE ប្រើលើកុំព្យូទ័រ ការបញ្ចូល library ប្រើជាមួយឧបករណ៍មួយចំនួន ជាពិសេសមានជាការសរសេរកូដ ជាមួយសៀវភៅត្រួតស្រាបៀរតែម្តង។ ដោយមូលដ្ឋានទាំងអស់ដែលបានលើកមកខាងលើយើងអាច យកទៅច្នៃប្រឌិតប្រើជាប្រយោជន៍បានច្រើនមុខទៀត ដូចជា ការបង្កើតកូនឡានបញ្ជា គ្រប់គ្រងការបិទបើកភ្លើងក្នុងផ្ទះ ការគ្រប់គ្រងប្រព័ន្ធស្រោចស្រពដំណាំ ក៏ដូចជាការឱ្យចំណីសត្វជាដើម។

គណៈកម្មការរៀបរៀងសង្ឃឹមថា សៀវភៅមួយក្បាលនេះ នឹងអាចជួយជាពន្លឺនាំផ្លូវអ្នកក្នុងការរៀនសរសេរកូដ Arduino ឱ្យបានល្អប្រសើរ និងមានជំនាញបន្ថែមជាអ្នក Programmer coding ហើយជាពិសេសអាចយកទៅប្រើការបានយ៉ាងពិតប្រាកដ។

សូមអរគុណ

កំពង់ឈើទាល ថ្ងៃទី២៣ ខែសីហា ឆ្នាំ២០២១

មាតិកា

ចំណងជើង

ទំព័រ

មេរៀនទី១ ស្វែងយល់ពី Arduino

១. ប្រវត្តិសង្ខេប Arduino	១
១.១ ប្រភេទរបស់ Board Arduino	២
ក. board ប្រភេទ Uno R3	២
ខ. board ប្រភេទ Uno SMD	២
គ. board ប្រភេទ Mega 2560 R3	២
ឃ. board ប្រភេទ DUE	៣
ង. board ប្រភេទ Leonardo	៤
ច. board ប្រភេទ Pro mini 328 3.3V	៤
ឆ. board ប្រភេទ Pro mini 328 ៥V	៤
១.២ អង្គប្រកបសំខាន់ៗរបស់ board Arduino uno	៤
២. Arduino IDE	៥
៣. <u>Menu សំខាន់ៗសម្រាប់ប្រើលើកដំបូង</u>	៥
ក. ការជ្រើសរើសប្រភេទ Board	៥
ខ. ការជ្រើសរើស Port	៦
គ. ការជ្រើសរើស Library	៦
ឃ. កាបង្ហាញ menu icon	៧
ង. ការបញ្ចូលកូដទៅក្នុង Board	៧

មេរៀនទី២ Digital Output

១. Blinking តាមរយៈ Arduino	៨
ក. អញ្ញាត (Variable)	៨
ខ. Sketch Overview	៨
គ. Pin Direction	៩
ឃ. ការបញ្ចេញសញ្ញាល់ឌីជីថល (Digital Write)	១០
ង. Delay Time	១០
ច. LED Blinking	១១
ឆ. ការប្រើប្រាស់អញ្ញាត (Variable)	១២
២. ប្រព័ន្ធលេខគោល	១៤

៣. ទំរង់នៃពាក្យបញ្ជា	១៥
ក. ពាក្យបញ្ជា For	១៥
ខ. ពាក្យបញ្ជា if	១៦
គ. ពាក្យបញ្ជា if-else	១៧
ឃ. ពាក្យបញ្ជា if...else if...else	១៩
ង. ពាក្យបញ្ជា while	២១
ច. ពាក្យបញ្ជា do...while	២១

មេរៀនទី៣ អង្គតំលេខ 7 Segment

១. 7 Segment 1digit	២៣
ក. ការប្រើ 7 Segment នឹងលេខគោល២	២៤
ខ. ការប្រើ 7 Segment នឹងលេខគោល១៦	២៨
២. 7 Segment 2digit	៣៣
ក. ការបង្កើត Function declaration	៣៧
៣. Driver for 7 Segment 2 More than digit (Max7219)	៣៨
ក. Shutdown Mode (0xXC):	៤២
ខ. Decode Mode (0xX9);	៤៣
គ. Scan Limit (0xBB);	៤៤
ឃ. Intensity (0xA);	៤៤
ង. Display Test (0xFF);	៤៥

មេរៀនទី៤ Digital Input

១. Comparison Operator	៤៩
២. Bitwise Operator	៤៩
៣. Boolean Operator	៥០
៤. ការគណនាបែបគណិតវិទ្យា	៥០
៥. Digital Input Arduino	៥១

មេរៀនទី ៥ Serial Monitoring និង String Function

១. Serial of Arduino	៥៥
២. Sting Objects	៦១

មេរៀនទី ៦ Display LCD, OLED

១. Liquid Crystal Display (LCD)	៧០
ក.ការប្រើ LCD I2C	៨១

២. Graphic OLED	៨៣
ក. ទម្រង់ដែលត្រូវប្រើជាមួយ OLED	៨៦
ខ. ដាក់រូបក្នុង OLED	៨៩
គ. ការប្រើ Real Time ជាមួយ OLED	៩៧
មេរៀនទី ៧ EEPROM និង SD Card	
១. KeyPap	១០៧
២. EEPROM	១១៤
ក. ពាក្យបញ្ជាដែលត្រូវប្រើជាមួយ EEPROM	១១៤
៣. SD Card	១១៩
ក. ការចាក់បទចំរៀងចេញពី SD card	១២៤
ខ. ការចាក់ភ្លេងដោយមិនប្រើ SD card	១២៦
មេរៀនទី ៨ Basic Wireless Communication	
១. IR Remote Control	១៣៤
ក. Receiver	១៣៤
ខ. Transmitter	១៣៦
២. UART Tx/Rx Tranceiver module (HC-12)	១៤៦
មេរៀនទី ៩ Basic Motor control	
១. Analog input/output	១៥៦
ក. Analog Read	១៥៦
ខ. Analog Write (Pule Width Modulation)	១៥៧
២. DC Motor Control	១៦១
៣. Servo Motor	១៦៨
មេរៀនទី ១០ Sensor	
១. Thermistor	១៧២
២. Distance Sensor	១៧៦
៣. Security Sensor	១៨០
ក. Motion Sensor	១៨០
ខ. Flame Sensor	១៨១
គ. Alcohol Sensor	១៨៣
ឃ. Gas Methan Sensor	១៨៤
៤. Agriculture Sensor	១៨៦

ក. Soil Sensor	១៨៦
ខ. Rain Sensor	១៨៧
៥. Switch Sensor	១៨៩
ក. Capacitive touch Switch	១៨៩

មេរៀនទី១ ស្វ័យយល់ពី Arduino

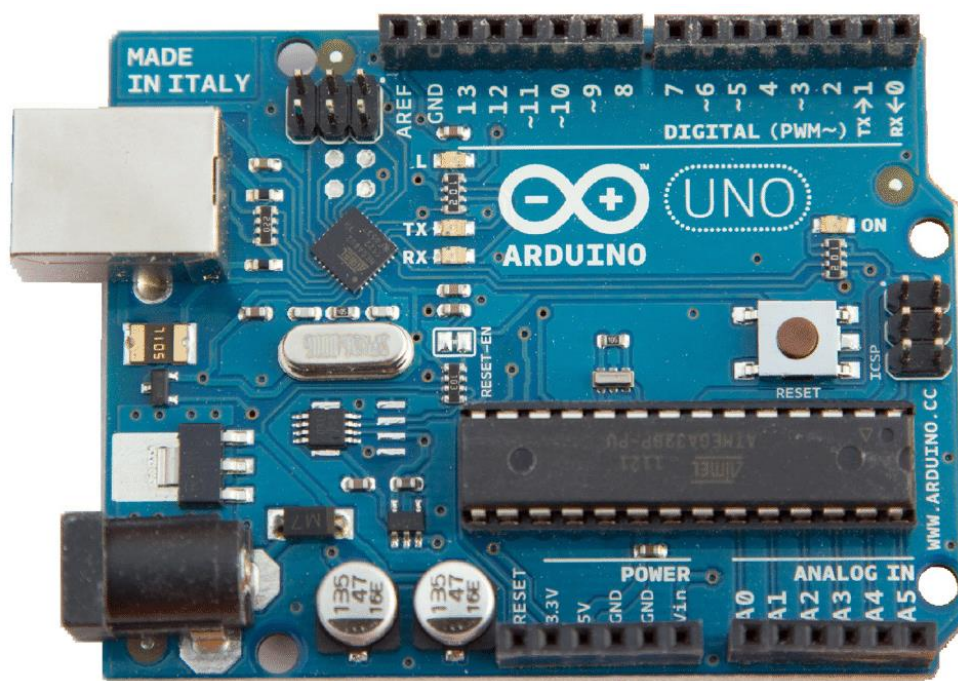
១. ប្រវត្តិសង្ខេប Arduino

Board Arduino ត្រូវបានលោក Massimo Banzi និងលោក Davi Cuartielles បង្កើតឡើងក្នុងឆ្នាំ ២០០៥ ក្នុងគោលបំណងធ្វើឱ្យមានតម្លៃថោក និងងាយស្រួលសម្រាប់មនុស្សទូទៅប្រើប្រាស់។ Arduino អានថា (អា ខូ នូ ឬ អា ខូយ នូ) ជា Board ដែលមានការអភិវឌ្ឍទាំងផ្នែក Software និង Hardware បែបបើកបង្ហាញទិន្នន័យ (Open Source) ដោយ Hardware មានលក្ខណៈងាយស្រួលក្នុងការប្រើ និងមានតម្លៃទាប។ ចំណែក Software ជាប្រភេទកូដដែលមានភាពងាយស្រួលក្នុងការស្វែងយល់ ដូចនេះហើយវាសមស្របជាមួយសិស្ស និងស្វ័យយល់កម្មកសិកម្មអភិវឌ្ឍន៍ចាប់ពី កម្រិតទាប រហូតដល់កម្រិតខ្ពស់។ តើ Arduino ជាអ្វី?

Arduino ជាបង្កើតសៀគ្វីរបស់ Microcontroller Atmel ដែលវាផ្តល់ឱ្យអ្នកប្រើប្រាស់ទូទៅមានភាពងាយស្រួល ក្នុងការតសៀគ្វីដោយពុំចាំបាច់មានមូលដ្ឋានគ្រឹះផ្នែកអេឡិចត្រូនិចពីមុនមក។ ក្នុងគេហទំព័ររបស់ Arduino មានផ្តល់នូវគំរូកូដ សៀគ្វីស្រាប់ និង Library សម្រាប់ប្រើជាមួយឧបករណ៍អេឡិចត្រូនិចជាច្រើន ដែលគេបង្កើតឡើង សម្រាប់ជាតម្រូវការភ្ជាប់ជាមួយ Arduino ស្រាប់ៗដោយពុំស្មុគស្មាញ។

តើអ្នកធ្លាប់សង្កេតមើលម៉ាស៊ីនបោកខោអាវ ដំណើរយន្ត គ្រឿងក្មេងលេង ឧបករណ៍អេឡិចត្រូនិចផ្សេងៗទេ? ហើយឆ្ងល់ទេថាវាដំណើរការបានយ៉ាងដូចម្តេច? តើអ្នកធ្លាប់មានគំនិតថាចង់បង្កើតវាដោយដៃខ្លួនឯងដែរឬទេ?

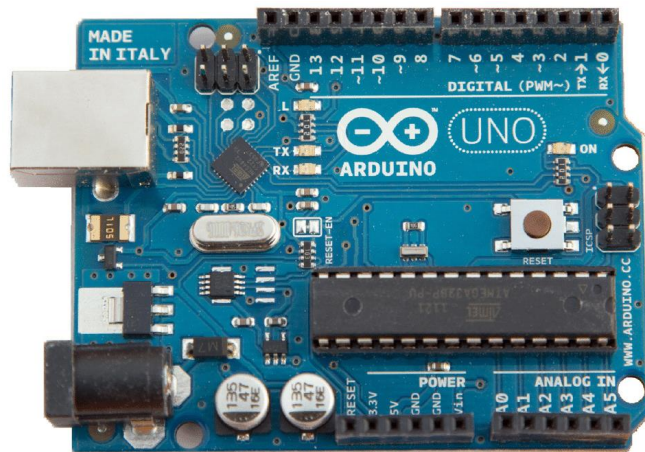
Board Arduino ក្នុងរូបទី១.១ អាចបកស្រាយចម្បងនេះបាន ហើយវាក៏នឹងមិនមែនជាអាថ៌កំបាំងអេឡិចត្រូនិចតទៅទៀតដែរ។



រូបទី១.១ Board Arduino Uno

១.១ ប្រភេទរបស់ Board Arduino

ក. board ប្រភេទ Uno R3 ជា Board Arduino ដែលទទួលបានការពេញនិយមច្រើនបំផុតក្នុងការប្រើប្រាស់ធ្វើ Project ផ្សេងៗ ដោយវាតូច តម្លៃទាប និងមាន Library ផ្សេងៗ ដែលបានអភិវឌ្ឍន៍ជាបន្តបន្ទាប់ដើម្បីគាំទ្រដល់ Board មួយនេះ។



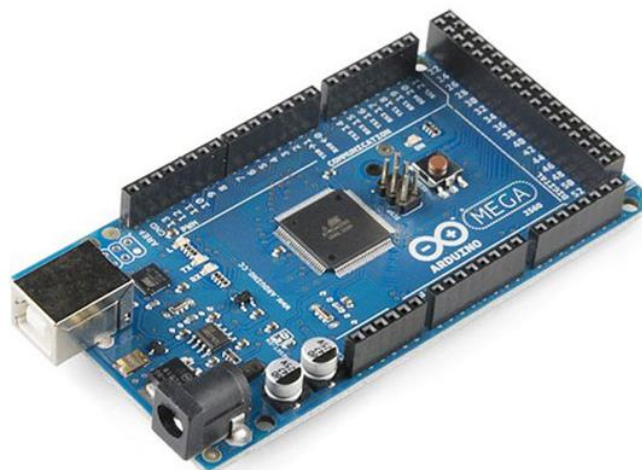
រូបទី១.២ Board Arduino Uno R3

ខ. board ប្រភេទ Uno SMD ជា Board ដែលមានលក្ខណៈសម្បត្តិនិងធ្វើការដូចគ្នាជាមួយ Board Arduino UNO R3 ដែរ តែខុសគ្នាត្រង់ Arduino SMD ប្រើមីក្រូខុនត្រូល័រ (Microcontroller) បែប Package SMD ចំណែកឯ Arduino UNO R3 ប្រើ microcontroller បែប Package DIP ។



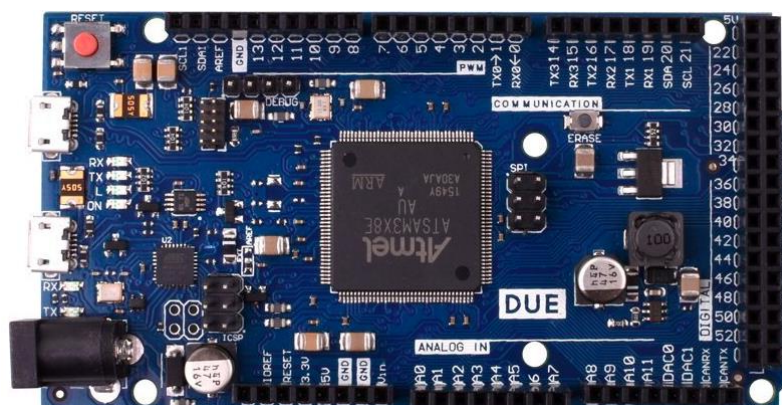
រូបទី១.៣ Board Arduino Uno SMD

គ. board ប្រភេទ Mega 2560 R3 ជាប្រភេទ Board ដែលបាន Design មានលក្ខណៈប្រហាក់ប្រហែលទៅនឹង Board Arduino UNO R3 ដែរ តែ input និង output របស់វាមានច្រើនជាង។ សម្រាប់ប្រើប្រាស់ វាជាមួយឧបករណ៍បានច្រើនប្រភេទជាមួយគ្នាក្នុងពេលតែមួយ និងមានការពេញនិយមប្រើប្រាស់ផងដែរក្នុងពេលបច្ចុប្បន្ន។



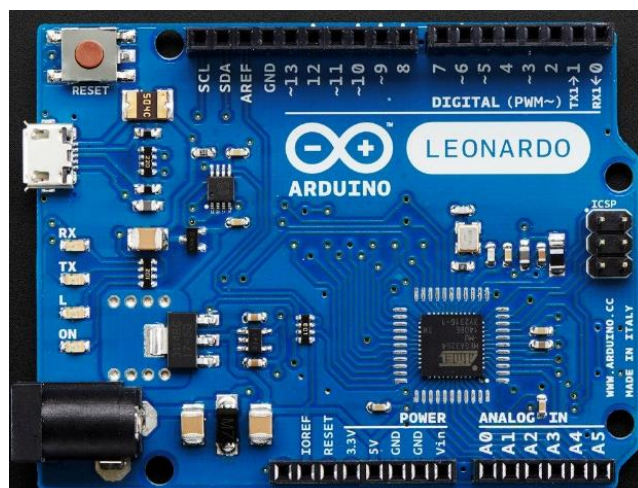
រូបទី១.៤ Board Arduino Mega 2560

ឃ. board ប្រភេទ DUE ជា Board Arduino ដែលមានឈ្មោះ microcontroller ថ្មី ដោយប្រើ AVRAT91SAM3X8E ជាពិសេសវាធ្វើការលឿនជាងមុន (លឿនជាង Mega 2560 R3) ។



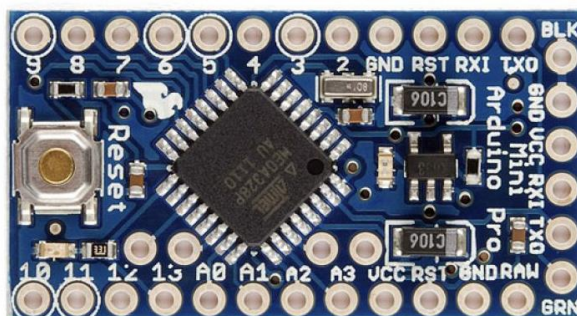
រូបទី១.៥ Board Arduino DUE

ង. board ប្រភេទ Leonardo ការប្រើប្រាស់ Board Leonardo មានលក្ខណៈស្រដៀងទៅនឹង UNO R3 តែប្រើបន្ទះឈ្មោះ microcontroller ប្រភេទ ATmega32U4 ដែលធ្វើឱ្យមាន USB តូចជាងមុន និង ត្រូវមាន Driver សម្រាប់USB ដែលខុសពី Board UNO R3 ។



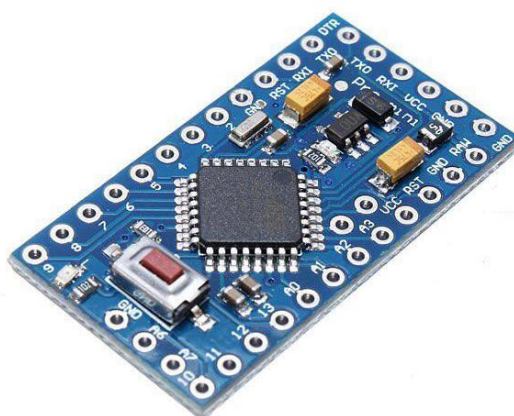
រូបទី១.៦ Board Arduino Leonardo

ប. board ប្រភេទ Pro mini 328 3.3V ជាប្រភេទ Board ដែលមានទំហំតូចប្រើបន្ទះ ឈ្មោះ microcontroller Atmega328 ហើយនៅលើ Board មានភ្លើងចិញ្ចឹមតែ ៣.៣V ប៉ុណ្ណោះ។ ចំណែកភ្លើងចេញពី Board ក៏មានត្រឹមតែ ៣.៣V ដែរ។



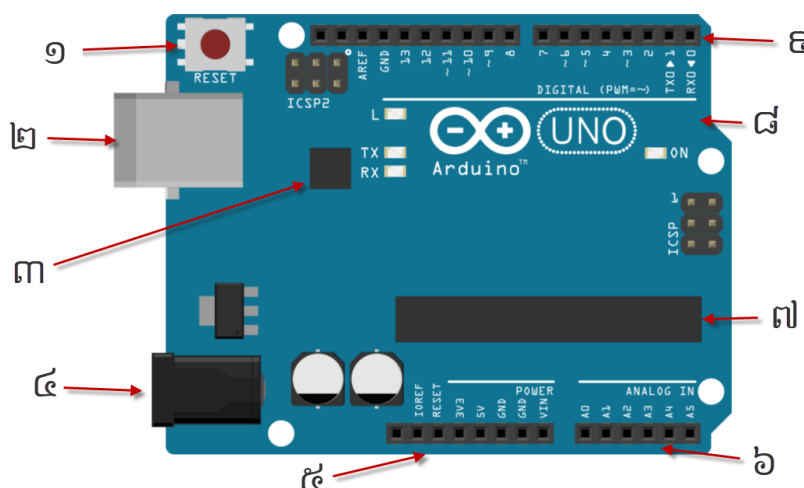
រូបទី១.៧ Board Arduino Pro mini 328 3.3V

ឆ. board ប្រភេទ Pro mini 328 5V ជាប្រភេទ Board ដែលមានទំហំតូចប្រើបន្ទះ ឈ្មោះ microcontroller Atmega328 ហើយនៅលើ Board មានភ្លើងចិញ្ចឹម 5V ។ ចំណែកភ្លើងចេញពី Board ក៏មាន 5V ដែរ។



រូបទី១.៨ Board Arduino Pro mini 328 3.3V

១.២ សមាសភាគសំខាន់ៗរបស់ board Arduino uno



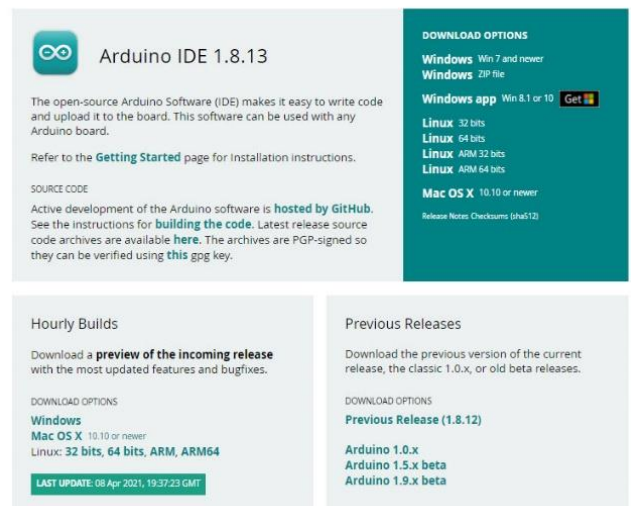
រូបទី១.៩ អង្គប្រកបសំខាន់ៗរបស់ board Arduino uno

1. ប៊ូតុងចុចសម្រាប់ reset
2. រន្ធសម្រាប់តឿន usb
3. Microcontroller ប្រភេទ atmega 16u2
4. រន្ធតភ្លើងបិញ្ចឹម dc 5v-12v
5. រន្ធតភ្លើងចេញ ៣.៣v; 5.5v និង ម៉ាស់ (gnd)
6. ជើងដោតខ្សែស៊ីញ៉ាល់ input Analog
7. Microcontroller atmega 328
8. ជើង icsp សម្រាប់usb interface
9. ជើងត input និង output digital

២. Arduino IDE

ដើម្បីអាចទាញយកកម្មវិធី Arduino IDE មកប្រើក្នុងកុំព្យូទ័របានយើងត្រូវចូលទៅកាន់គេហទំព័ររបស់ Arduino ផ្ទាល់ដែលមានឈ្មោះថា [https:// www.Arduino.cc/en/Main/Software](https://www.Arduino.cc/en/Main/Software) ការទាញយកមាន 2 របៀបគឺការទាញយកដោយដំឡើងក្នុងគ្រឿងតែម្តង និងការទាញយកជា file Program ហើយដំឡើងតាមក្រោយ ហើយយើងអាច copy file Program នេះទៅដំឡើងនៅលើកុំព្យូទ័រផ្សេងៗ ទៀតបាន។ ក្រោយពីយើងបានចូលក្នុង គេហទំព័រ ខាងលើហើយ យើងត្រូវចុចលើពាក្យ Windows Install > JUST DOWNLOAD ក្រោយមកវានឹងចេញផ្ទាំងប្តីឱ្យយើងចុច save បន្ទាប់មកពេល save ចប់យើងនឹងបាន File Program Arduino មួយ។

Downloads



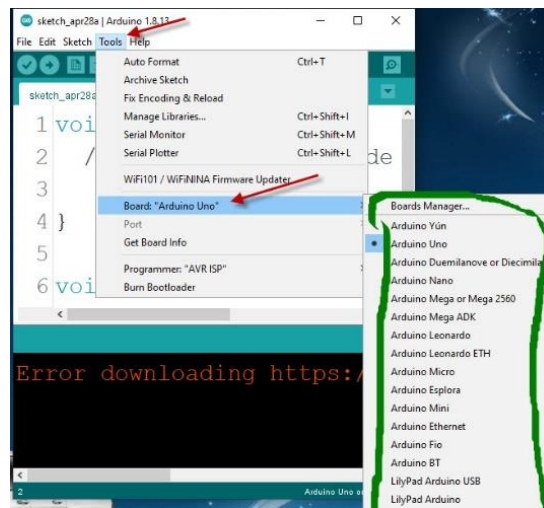
រូបទី១.១០ ទំព័រទាញយកកម្មវិធី Arduino IDE

វិធីដំឡើង ចុចពីរដងលើ File program > I Agree ដើម្បីដំឡើង បន្ទាប់មកចុច Next ហើយចុច Install រង់ចាំវាដំឡើងរួច វានឹងលោតឱ្យយើងដំឡើង Device សម្រាប់ភ្ជាប់ជាមួយ Board USB ហើយចុច Install (បើវាលោតឱ្យដំឡើង Device ច្រើនដង យើងក៏ចុច Install ច្រើនដងតាមនឹងដែរ) បន្ទាប់ពីរង់ចាំវាដំណើរការចប់យើងត្រូវចុចពាក្យ Close ជាការស្រេច។ ពេលនេះយើងមានកម្មវិធី Arduino ក្នុងកុំព្យូទ័រហើយ។

៣. Menu សំខាន់ៗសម្រាប់ប្រើលើកដំបូង

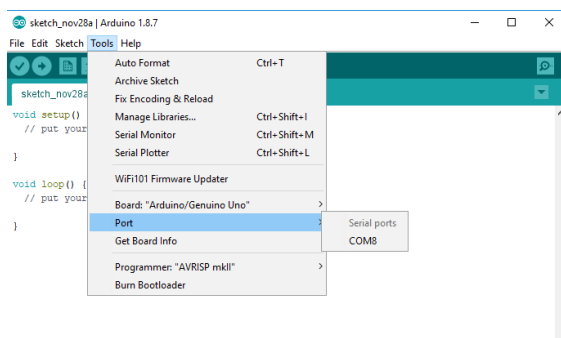
ក. ការជ្រើសរើសប្រភេទ Board

ការជ្រើសរើសប្រភេទ Board ចុច Tools > Board ពេលនេះយើងនឹងឃើញប្រភេទ Board ផ្សេងៗជាច្រើនឱ្យយើងរើស។ យើងត្រូវជ្រើសរើសប្រភេទ Board ឱ្យត្រូវជាមួយនឹង Board ដែលយើងកំពុងប្រើ។



រូបទី១.១១ ការជ្រើសរើស Board

ខ. ការជ្រើសរើស Port



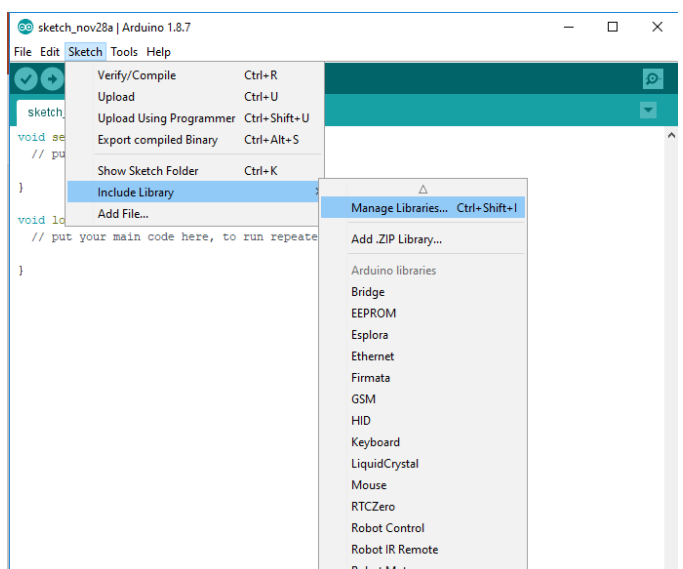
រូបទី១.១២ ការជ្រើសរើស Port

ដើម្បីជ្រើសរើស Port យើងត្រូវចុច Tools >

Port ពេលនេះនឹងមាន Port USB ឱ្យជ្រើសរើស។ យើងត្រូវជ្រើសរើសយក Port ដែលយើងបានដោត Board Arduino (COM1 ឬ COM2 ឬ ...)។

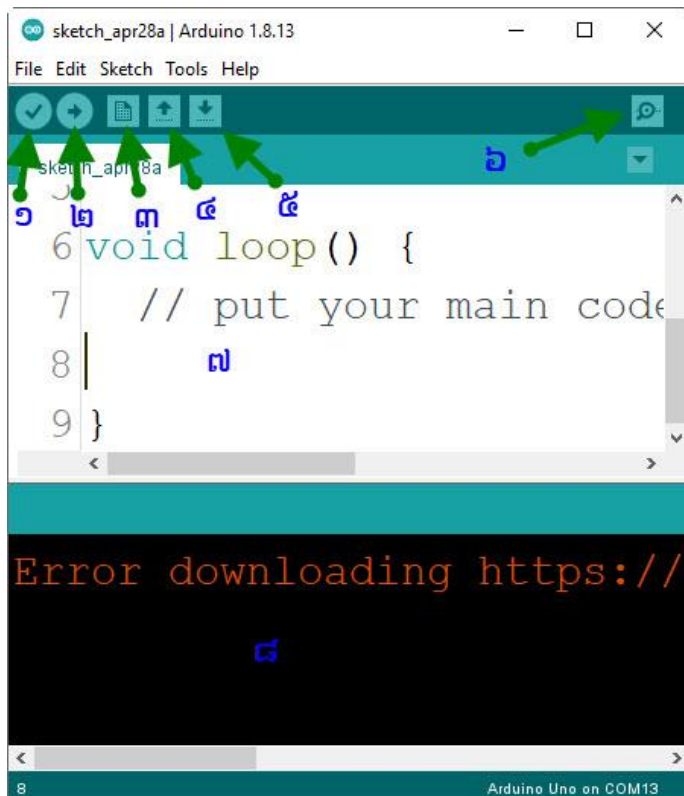
គ. ការជ្រើសរើស Library

ការជ្រើសរើស Library ចុច Sketch > Include Library ពេលនោះយើងនឹងឃើញ Library ជាច្រើនឱ្យជ្រើសរើស។ បើ Library មិនមានតាមដែលយើងចង់បាន យើងអាចទាញយក Library បន្ថែមទៀតបានដោយចុចលើពាក្យ Sketch > Include Library > Add.ZIP Library (ករណីយើងមាន Library ដែលយើងត្រូវការនៅក្នុងគ្រឿងកុំព្យូទ័រស្រាប់) ឬយើងអាច Download Library បន្ថែមបានដូចគ្នាដែរដោយត្រូវចុច Sketch > Include Library > Manage Library (ដោយយើងត្រូវតភ្ជាប់ Internet ជាមុនសិន)។



រូបទី១.១៣ ការជ្រើសរើស Library

២២. ការបង្ហាញ menu icon



រូបទី១.១៤ ការបង្ហាញ menu icon

១. Verify: មានតួនាទីត្រួតពិនិត្យភាពត្រឹមត្រូវរបស់កូដដែលយើងបានសរសេរ។

២. Upload: មានតួនាទីបញ្ចូលកូដទៅក្នុង Board Arduino។

៣. New Tab: មានតួនាទីបើកទំព័រសរសេរកូដថ្មី (បើក Arduino IDE ថ្មីមួយទៀត)។

៤. Open: មានតួនាទីទាញយកកូដដែលបានរក្សាទុក។

៥. Save: មានតួនាទីរក្សាកូដទុកក្នុងកុំព្យូទ័រ។

៦. Serial Monitor: មានតួនាទីបើកមើលផលការធ្វើការរបស់កូដដែលយើងបានកំណត់។

៧. Code goes here: គ្រប់កូដទាំងអស់ត្រូវសរសេរនៅទីនេះ។

៨. Console: មានតួនាទីបង្ហាញពីភាពត្រឹមត្រូវនិងកំហុសផ្សេងៗដែលកើតមានឡើងនៅលើកូដ។

១. ការបញ្ចូលកូដទៅក្នុង Board

ដើម្បីបញ្ចូលកូដ មីក្រូខនត្រូល័រ (Microcontroller) ដែលយើងសរសេរហើយទៅក្នុង Board Arduino យើងធ្វើបានតាមជំហានទាំង៣ ដូចខាងក្រោម៖

ជំហានទី១. ជ្រើសរើស Board សូមចុច Tools > Board > យកឱ្យត្រូវនឹង Board ដែលយើងកំពុងប្រើ។

ជំហានទី២. ជ្រើសរើស Port សូមចុច Tools > Port > COM១, COM២... យើងរើសតាមភាពជាក់ស្តែង ដោយភាគច្រើន Port ដែលយើងកំពុងប្រើគឺនៅខាងក្រោមគេ។

ជំហានទី៣. Upload យើងគ្រាន់តែចុចលើមីនីស្ត្រូបូត Upload បន្ទាប់មកយើងត្រូវចាំរហូតដល់ចេញពាក្យ Done Uploading គឺមានន័យថាការបញ្ចូលកូដបានសម្រេច។

មេរៀនទី២ Digital Output

១. Blinking តាមរយៈ Arduino

យើងធ្លាប់ស្គាល់ IC 555 រួចមកហើយ គឺវាមានភាពលំបាកខ្លាំងណាស់នៅពេលដែលយើងត្រូវការប្តូរប្រេកង់របស់វាម្តងៗ ដោយសារតែ Hardware ម៉ូដែលចាស់ៗ។ តទៅនេះយើងសិក្សាអំពី Programming និង Coding។

ក. អញ្ញាត (Variable) អញ្ញាតនៅទីនេះមានន័យថា អក្សរសម្គាល់ឱ្យចំនួនលេខឬចំនួនតួអក្សរដែលយើងសរសេរកូដ។

តារាងតម្លៃ នៃការប្រើអញ្ញាតនីមួយៗ

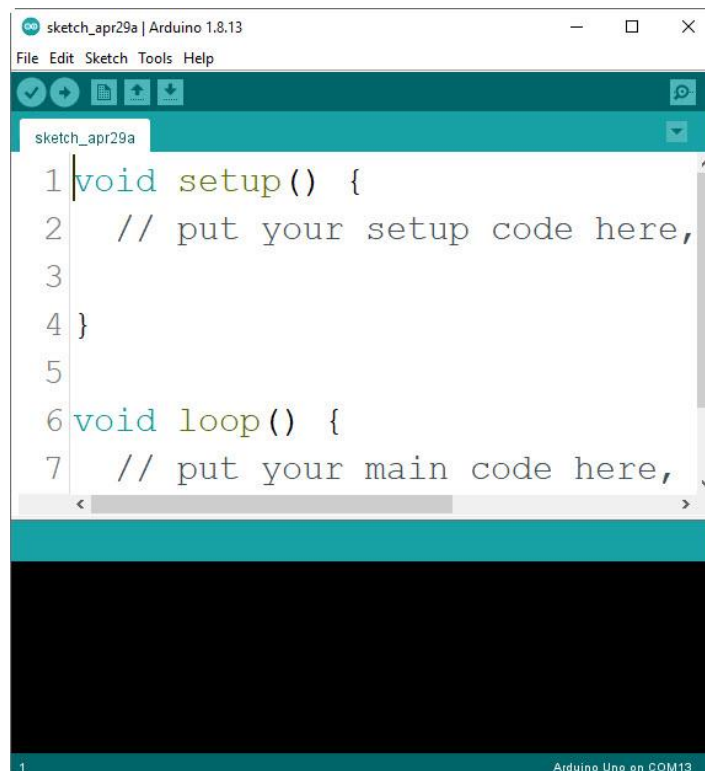
ប្រភេទអញ្ញាត	ចំនួន Byte	តម្លៃ	ការប្រើប្រាស់
int	2	-32768ទៅ32767	ជាលេខចំនួនគត់ដែលមានតម្លៃដូចខាងឆ្វេង
Unsigned int	2	0ទៅ65535	ជាលេខចំនួនគត់ដែលមិនមានតម្លៃអវិជ្ជមាន
long	4	-2147483648 ទៅ2147483647	ជាលេខចំនួនគត់ដែលមានតម្លៃធំជាង int
Unsigned long	4	0ទៅ65535	ជាលេខចំនួនគត់ដែលមានតម្លៃធំជាង int និងមិនមានតម្លៃអវិជ្ជមាន
float	4	-3.4028235E+38ទៅ 3.4028235E+38	សម្រាប់ប្រើលើតម្លៃទសភាគ
double	4	-3.4028235E+38ទៅ 3.4028235E+38	ប្រើលើតម្លៃទសភាគតែមានភាពលម្អៀងតិច
byte	1	0ទៅ255	ជាលេខចំនួនតម្លៃតិចជាង 255
boolean	1	ពិត (1) និងមិនពិត (0)	តម្លៃតក្កវិទ្យា

២. Sketch Overview

នៅពេលដែលយើងចាប់ផ្តើមបើកកម្មវិធី Arduino ភ្លាម នោះយើងនឹងឃើញពីរតំបន់ខុសគ្នា ដូចក្នុងរូបទី២.១ ដែលជាកន្លែងសម្រាប់សរសេរកូដ។ ទម្រង់នៃការសរសេរកូដ Arduino មានតែ ២ ផ្នែកសំខាន់ៗគឺ៖

Void setup (): ជាផ្នែកកំណត់តម្លៃដែលជាពាក្យបញ្ជានៃការចាប់ផ្តើមធ្វើការរបស់កម្មវិធី ហើយនៅផ្នែក Void setup នេះវាធ្វើការតែម្តងគត់។

Void loop (): ផ្នែក Loop ជាផ្នែកសម្រាប់សរសេរកូដរួមទាំងអស់ ហើយផ្នែកនេះធ្វើការវិលទៅវិលមក គឺពេលធ្វើការចប់ពាក្យបញ្ជាឬកូដទាំងអស់វានឹងវិលមកចាប់ផ្តើមធ្វើការសារថ្មីម្តងទៀត។



រូបទី២.១ ផ្ទាំងកម្មវិធី Arduino IDE

គ. Pin Direction

ដោយសារតែជើង Digital របស់ Arduino មានទិសដៅពីរ ពោលគឺទិសដៅចេញ និងទិសដៅចូល ដូច្នេះហើយមុននឹងប្រើជើងនេះយើងត្រូវប្រកាសទិសដៅរបស់វាជាមុនសិន។

ទម្រង់ការប្រើ **pinMode** ៖

pinMode(pin, mode);

ដែលPin: លេខជើងឌីជីថល (unsigned int)

Mode: ចែកចេញជាបីគឺ **OUTPUT**, **INPUT**, **INPUT_PULLUP**។

Mode នីមួយៗត្រូវបានគេប្រើដើម្បី៖

OUTPUT: ប្រកាសឱ្យជើងណាមួយជាជើងចេញស៊ីញ៉ាល់ឌីជីថល

INPUT: ប្រកាសឱ្យជើងណាមួយជាជើងចូលស៊ីញ៉ាល់ឌីជីថល

INPUT-PULLUP: ប្រកាសឱ្យជើងណាមួយជាជើងចូលតែមានរស៊ីស្តង់ Pull Up ពីខាងក្នុងហើយស្រេច។

ឧទាហរណ៍

```
void setup( )
{
  pinMode(9, OUTPUT);
}
```

```
void loop( )
```

```
{
}
```

ឃ. ការបញ្ជូនស៊ីញ៉ាល់ឌីជីថល (digitalWrite)

របៀបដែល Arduino បញ្ជូនស៊ីញ៉ាល់ទៅក្រៅដើម្បីបញ្ជាទៅកាន់អ្វីមួយត្រូវបានហៅតាមរយៈទម្រង់ខាងក្រោម៖

```
ទម្រង់ digitalWrite( );
digitalWrite( pin, value );
```

ដោយ pin: ជើងដែលត្រូវបញ្ជូន (unsigned int)

Value: តម្លៃជាឌីជីថលដែលត្រូវចេញ ពិត (1) មិនពិត (0) ឬក៏ HIGH, LOW

ឧទាហរណ៍

```
void setup( )
```

```
{
```

```
  pinMode(7, OUTPUT);
```

```
}
```

```
void loop( )
```

```
{
```

```
  digitalWrite( 7, HIGH );
```

```
}
```

ង. Delay Time

ការពន្យារពេលនៃព្រឹត្តិការណ៍ណាមួយដែលកំពុងកើតឡើងនៅក្នុងកូដរបស់យើង។ នៅក្នុង Arduino IDE, delay ចែកចេញជាពីរទម្រង់

```
ទម្រង់ delay( );
```

- delay(ms);
- delayMicroseconds(us);

ដោយ ms: រយៈពេលគិតជាមីលីវិនាទី (unsigned int)

us: រយៈពេលគិតជាមីក្រូវិនាទី (unsigned int)

ឧទាហរណ៍

```
void setup( )
```

```

{
  pinMode(7, OUTPUT);
}
void loop( )
{
  digitalWrite(7, HIGH);
  delay(1000);
}

```

៥. LED Blinking

យើងធ្លាប់សិក្សារួចមកហើយពីរបៀបនៃការធ្វើ LED Blinking តាមរយៈ IC 555 ដែលវាមានភាពលំបាកក្នុងការគណនា និងត្រូវផ្លាស់ប្តូរ Hardware រហូត រាល់ពេលដែលយើងត្រូវការដូចប្រភេទឬរយៈពេល ON/ OFF LED របស់វាម្តងៗ។ តែសម្រាប់ Arduino វិញយើងអាចដោះស្រាយការលំបាកនេះបាន ដោយការសរសេរកូដដូចក្នុងឧទាហរណ៍ខាងក្រោម៖

ឧទាហរណ៍១ សរសេរកូដបញ្ជា LED ១គ្រាប់ឱ្យភ្លឺហើយរលត់ទៅវិញក្នុងរយៈពេល ១វិនាទី។

```

void setup( )
{
  pinMode(7, OUTPUT);
}
void loop( )
{
  digitalWrite(7, HIGH);
  delay(1000);
  digitalWrite(7, LOW);
}

```

ឧទាហរណ៍២ សរសេរកូដបញ្ជា LED ៥គ្រាប់ឱ្យភ្លឺហើយរលត់ទៅវិញ ក្នុងរយៈពេល ១វិនាទី ដោយភ្លឺរលត់ម្តងមួយៗក្នុងរយៈពេលដូចគ្នាទាំងអស់។

```

void setup( )
{
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);

```

```

    pinMode( 7, OUTPUT );
    pinMode( 8, OUTPUT );
    pinMode( 9, OUTPUT );
}
void loop( )
{
    digitalWrite( 5, HIGH );
    delay( 1000 );
    digitalWrite( 5, LOW );
    delay( 1000 );
    digitalWrite( 6, HIGH );
    delay( 1000 );
    digitalWrite( 6, LOW );
    delay( 1000 );
    digitalWrite( 7, HIGH );
    delay( 1000 );
    digitalWrite( 7, LOW );
    delay( 1000 );
    digitalWrite( 8, HIGH );
    delay( 1000 );
    digitalWrite( 8, LOW );
    delay( 1000 );
    digitalWrite( 9, HIGH );
    delay( 1000 );
    digitalWrite( 9, LOW );
    delay( 1000 );
}

```

ន. ការប្រើប្រាស់អញ្ញាត (Variable)

អញ្ញាតត្រូវបានប្រកាសឱ្យប្រើប្រាស់តាមរយៈការដាក់ប្រភេទទិន្នន័យរបស់វានៅពីខាងមុខរួចដាក់ឈ្មោះ ហើយឈ្មោះនោះ នឹងអាចត្រូវបានហៅឱ្យធ្វើការគ្រប់ពេលក្នុងកូដរបស់យើង។ អញ្ញាតចែកជាពីរ គឺ អញ្ញាតទាំងឡាយណាដែលប្រកាសនៅពីខាងលើ Void setup() ហៅថា Global Variable ឯអញ្ញាតដែលប្រកាសភ្លាមៗសម្រាប់តែប្រើក្នុងកន្លែងណាមួយ ហៅថា Temporary Variable ។

ឧទាហរណ៍១

```
int pinout = 7;
void setup( )
{
    pinMode(7, OUTPUT);
}
void loop( )
{
    digitalWrite( pinout, HIGH );
    delay( 1000 );
}
```

ឧទាហរណ៍២ សរសេរកូដបញ្ជា LED ៥គ្រាប់ឱ្យភ្លឺហើយរលត់ទៅវិញ ក្នុងរយៈពេល ១វិនាទី ដោយភ្លឺរលត់ម្តងមួយៗក្នុងរយៈពេលដូចគ្នាទាំងអស់។

```
int LED1 = 5;
int LED2= 6;
int LED3 = 7;
int LED4 = 8;
int LED5 = 9;
void setup( )
{
    pinMode( LED1, OUTPUT );
    pinMode( LED2, OUTPUT );
    pinMode( LED3, OUTPUT );
    pinMode( LED4, OUTPUT );
    pinMode( LED5, OUTPUT );
}
void loop( )
{
    digitalWrite( LED1, HIGH );
    delay( 1000 );
    digitalWrite( LED1, LOW );
    delay( 1000 );
    digitalWrite( LED2, HIGH );
```

```

delay( 1000 );
digitalWrite( LED2, LOW );
delay( 1000 );
digitalWrite( LED3, HIGH );
delay( 1000 );
digitalWrite( LED3, LOW );
delay( 1000 );
digitalWrite( LED4, HIGH );
delay( 1000 );
digitalWrite( LED4, LOW );
delay( 1000 );
digitalWrite( LED5, HIGH );
delay( 1000 );
digitalWrite( LED5, LOW );
delay( 1000 );
}

```

អនុវត្ត១

ចូរអ្នកត LED ឱ្យបាន 12 គ្រាប់ រួចសរសេរកូដដើម្បីឱ្យវាអាចបញ្ចេញពន្លឺ LED រត់ប្រដេញបណ្តាក់គ្នាជាខ្សែពីមួយទៅមួយ។

អនុវត្ត២

សរសេរកូដបង្កើតសិល្បៈ នៃការលោតក្តឹបភ្លេចឬប្រដេញគ្នា ធ្វើយ៉ាងណាឱ្យបានស្អាតនិងទាក់ទាញ (Cute)។

២. ប្រព័ន្ធលេខគោល

នៅក្នុងការសរសេរកូដ មានប្រព័ន្ធលេខគោលចំនួនបីដែលគេពេញនិយមយកមកប្រើប្រើនជាងគេនោះគឺ លេខគោល២ (Binary Number), លេខគោល១០ (Decimal Number) និងលេខគោល១៦ (Hexadecimal Number)។ នៅក្នុង Arduino លេខទាំងនោះត្រូវបានគេសរសេរកូដទម្រង់ខុសៗគ្នា។

ទម្រង់ការប្រើលេខគោល

Byte X = B00001011,	// binary number of X =11
Byte X = 11,	// decimal number of X =11
Byte X = 0x0b	// hexadecimal number of X = 11

តារាងប្រៀបធៀបលេខគោល

លេខគោល២	លេខគោល១០	លេខគោល១៦
B0000 1011	11	0x0B
B0001 0001	17	0x11
B0011 0010	50	0x32
B0010 0000 0000	512	0x200
B0111 1111	127	0x7F

៣. ទម្រង់នៃពាក្យបញ្ជា

ក. ពាក្យបញ្ជា for

for ជាពាក្យបញ្ជាធ្វើការវិលទៅវិលមក ដើម្បីត្រួតពិនិត្យលក្ខខណ្ឌដែលយើងកំណត់ ហើយវាវិលទៅវិលមករហូតដល់វាមិនពិត។ ហើយពាក្យបញ្ជា for ជាកន្លែងការងារមួយដែលធ្វើឡើងដដែលៗក្នុងចំនួនកំណត់ណាមួយ ដែលចំនួននេះត្រូវបានគេរាប់ចុះ ឬរាប់ឡើង។

ទម្រង់នៃពាក្យបញ្ជា for

For (initialization, condition, increment)

{

// statement(s)

}

for (លក្ខខណ្ឌចាប់ផ្តើម, លក្ខខណ្ឌចុងក្រោយ, ការធ្វើការរបស់លក្ខខណ្ឌ)

{

// សរសេរកូដ

}

រូបខាងក្រោមបង្ហាញអំពីដំណើរការដែល for ធ្វើការ

Parenthesis

Declare variable (optional)

Initialize

test

increment or

decrement

For (int x = 0, x < 100, x++)

{

println(x), // prints 0 to 99 }

ឧទាហរណ៍

```

void setup ( )
{
    pinMode ( 7,OUTPUT );
    pinMode ( 9, OUTPUT );
    for( int x = 0; x <10;x++ )
    {
        digitalWrite ( 7, HIGH );
        delay ( 500 );
        digitalWrite ( 7, LOW );
        delay ( 500 );
    }
}

void loop ( ) {
    digitalWrite ( 9, HIGH );
    delay ( 100 );
    digitalWrite ( 9, LOW );
    delay ( 100 );
}

```

ខ. ពាក្យបញ្ជា if

ពាក្យបញ្ជា if ជាពាក្យបញ្ជាដែលប្រើសម្រាប់កំណត់លក្ខខណ្ឌដោយយកតាមគោលការណ៍របស់គណិតវិទ្យាមកជួយ ។

ទម្រង់នៃពាក្យបញ្ជា if

```

if ( លក្ខខណ្ឌ )
{
    // ពាក្យបញ្ជា ឬកូដផ្សេងៗ
}

```

ឧទាហរណ៍

```
int R = 2;
```

```

int LED = 3;
void setup( )
{
    pinMode(R, INPUT);    // កំណត់ឱ្យអថេរ R ជា INPUT
    pinMode(LED, OUTPUT); // កំណត់ដើង LED ជា OUTPUT
}
void loop( )
{
    if(R == HIGH)          // ត្រួតពិនិត្យលក្ខខណ្ឌប្រសិនបើអថេរ R មានតម្លៃ HIGH
    {
        digitalWrite(LED,HIGH) // ឱ្យដើង LED មានតម្លៃ HIGH
    }
}

```

គ. ពាក្យបញ្ជា if-else

ពាក្យបញ្ជា if-else ជាពាក្យបញ្ជាដែលប្រើសម្រាប់កំណត់លក្ខខណ្ឌនៃការជ្រើសរើស ដោយអាចជ្រើសរើសលក្ខខណ្ឌតាមដែលត្រូវការ។

ទម្រង់នៃពាក្យបញ្ជា if-else

```

if( លក្ខខណ្ឌ )
{ // ពាក្យបញ្ជា ឬកូដផ្សេងៗ }
else
{ // ពាក្យបញ្ជា ឬកូដផ្សេងៗ }

```

ឧទាហរណ៍១

```

void setup( ) {
    pinMode(3, INPUT);    // កំណត់ដើងទី 3 ជា INPUT
    pinMode(4, OUTPUT);   // កំណត់ដើងទី 4 ជា OUTPUT
    pinMode(5, OUTPUT);   // កំណត់ដើងទី 5 ជា OUTPUT
}
void loop( ) {
    if( digitalRead(3) == HIGH);    // ដាក់លក្ខខណ្ឌឱ្យដើងទី 3 មានតម្លៃស្មើ HIGH
    {
        digitalWrite(4,HIGH);    // កំណត់ឱ្យដើងទី 4 មានតម្លៃ HIGH
    }
}

```

```

        digitalWrite(5,LOW);    // កំណត់ឱ្យជើងទី 5 មានតម្លៃ LOW
    }
    else                          // ប្រសិនបើផ្ទុយពីលក្ខខណ្ឌខាងលើ
    {
        digitalWrite(5, HIGH); // កំណត់ឱ្យជើងទី 5 មានតម្លៃ HIGH
        digitalWrite(4, LOW);  // កំណត់ឱ្យជើងទី 4 មានតម្លៃ LOW
    }
    delay(1000);                // ធ្វើការរយៈពេល ១ វិនាទី
}

```

ឧទាហរណ៍២ ប្រើនិមិត្តសញ្ញា && (និង)

```

void setup() {
    pinMode(3,OUTPUT);
    pinMode(2,INPUT);
    pinMode(1,INPUT);
}

void loop() {
    if((digitalRead(1)>0) && (digitalRead(2)>0))
    {
        digitalWrite(3,HIGH);
    }
    else                          // ប្រសិនបើផ្ទុយពីលក្ខខណ្ឌខាងលើ
    {
        digitalWrite(3,LOW);
    }
    delay(1000);
}

```

ឧទាហរណ៍៣ ប្រើនិមិត្តសញ្ញា || (ឬ)

```

void setup() {
    pinMode(3,INPUT);
    pinMode(4,INPUT);
    pinMode(5,OUTPUT);
}

```

```

}
void loop( ) {
    if( ( digitalRead( 3 )>0 ) || ( digitalRead( 4 )>0 ) )
    {
        digitalWrite( 5,HIGH );
    }
else                                     // ប្រសិនបើផ្ទុយពីលក្ខខណ្ឌខាងលើ
{
        digitalWrite( 5,LOW );
    }
    delay( 1000 );
}

```

❖ សំនួរ តើ if និង if - else ខុសគ្នាត្រង់ណា ?

២២. ពាក្យបញ្ជា if...else if...else

ពាក្យបញ្ជា if...else if...else ជាពាក្យបញ្ជាសម្រាប់ត្រួតពិនិត្យលក្ខខណ្ឌ ២ ផ្នែក ដោយក្នុងមួយផ្នែកធ្វើការដូចពាក្យបញ្ជា if...else។ ហើយមានផ្នែកផ្សេងទៀតយើងអាចដាក់លក្ខខណ្ឌបានយ៉ាងច្រើន លក្ខខណ្ឌទៅតាមតម្រូវការដែលត្រូវប្រើ ដោយលក្ខខណ្ឌទាំងនេះមិនជាន់គ្នា។

ទម្រង់ if...else if...else

```

if( លក្ខខណ្ឌទី១ )
{
    // កូដផ្សេងៗទី១
} else if( លក្ខខណ្ឌទី២ )
{
    // កូដផ្សេងៗទី២
} else
{
    // កូដផ្សេងៗទី៣
}

```

ឧទាហរណ៍

```

int S1 = 2;
int S2 = 3;
int M1 = 8;
int M2 = 9;
int M3 = 10;

void setup( ) {
    pinMode( S1, INPUT );           // កំណត់ជើងទី 2 ជាជើងចូល
    pinMode( S2, INPUT );           // កំណត់ជើងទី 3 ជាជើងចូល
    pinMode( M1, OUTPUT );          // កំណត់ជើងទី 8 ជាជើងចេញ
    pinMode( M2, OUTPUT );          // កំណត់ជើងទី 9 ជាជើងចេញ
    pinMode( M3, OUTPUT );          // កំណត់ជើងទី 10 ជាជើងចេញ
}

void loop( ) {
    if( digitalRead( S1 ) == HIGH ) // ដាក់លក្ខខណ្ឌឱ្យ S1 មានតម្លៃស្មើ HIGH
    {
        digitalWrite( M1, HIGH ); // បញ្ជាឱ្យ M1 ជា HIGH
    }
    else if( digitalRead( S2 ) == HIGH ) // ដាក់លក្ខខណ្ឌឱ្យ S2 មានតម្លៃស្មើ HIGH
    {
        digitalWrite( M2, HIGH ); // បញ្ជាឱ្យ M2 ជា HIGH
    }
    else // ប្រសិនបើខុសពីលក្ខខណ្ឌខាងលើ
    {
        digitalWrite( M3, HIGH ); // បញ្ជាឱ្យ M3 ជា HIGH
    }
    delay( 1000 );
}

```

ខ. ពាក្យបញ្ជា while

ពាក្យបញ្ជា while ជាពាក្យបញ្ជាដែលប្រើក្នុងការកំណត់លក្ខខណ្ឌ និងត្រួតពិនិត្យលក្ខខណ្ឌការធ្វើការថា តើ ទៅតាមអ្វីដែលយើងកំណត់ឬអត់ ដោយវាពិនិត្យលក្ខខណ្ឌ ហើយទើបធ្វើការតាមក្រោយ។

ទម្រង់ while

```
while ( លក្ខខណ្ឌ )
{
    // សរសេរកូដ
    លក្ខខណ្ឌ
}
```

ឧទាហរណ៍

```
int a;
void setup( ) {
    Serial.begin( 9600 );
}
void loop( ){
    a = 0;
    while ( a < 10 )
    {
        Serial.println( a );
        a++;
    }
}
```

ប. ពាក្យបញ្ជា do...while

ពាក្យបញ្ជា do...while ជាពាក្យបញ្ជាដែលធ្វើការមុនធ្វើតាមលក្ខខណ្ឌ ដោយធ្វើការតាមការសរសេរកូដទី១ (ទី១ក្នុងទម្រង់ធ្វើការ) ពេលត្រួតឃើញថាកូដទី១មិនពិត ឱ្យធ្វើការតាមកូដទី២។

ទម្រង់នៃការធ្វើការ do...while

```
do {
    // សរសេរកូដទី១ }
    while ( លក្ខខណ្ឌ )
    {
        // សរសេរកូដទី២ }
```

ឧទាហរណ៍

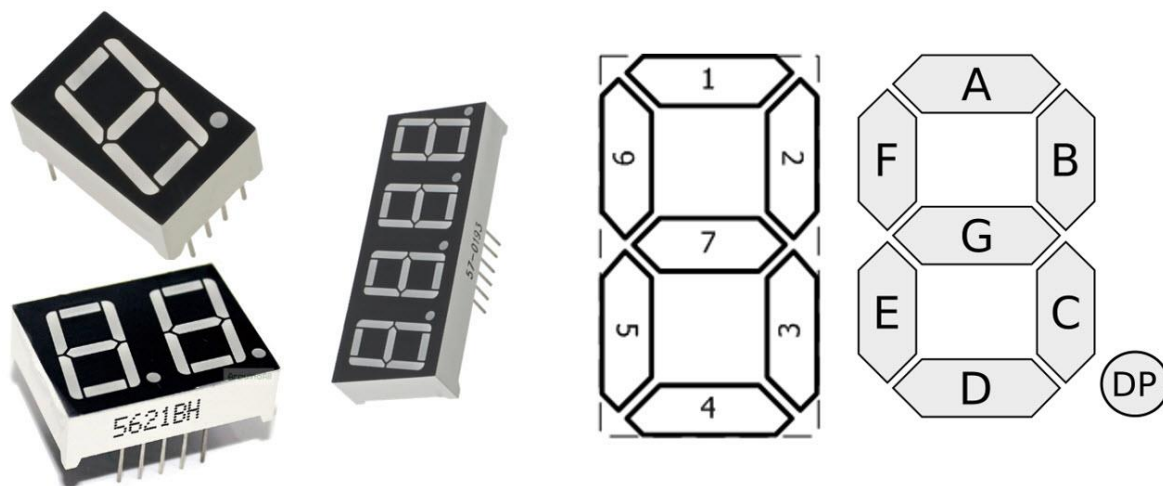
```

int i = 0;
void setup( )
{
    Serial.begin( 9600 );
}
void loop( )
{
    do {
        if ( i == 1 )
        {
            Serial.println( i );
        }
        else if( i > 1 )
        {
            Serial.println( "Good" );
        }
        i++;
    } while ( i<=3 );
}

```

មេរៀនទី៣ អង្គភាព 7 Segment

7 Segment ជាឧបករណ៍សម្រាប់បង្ហាញទិន្នន័យតួលេខ និងតួអក្សរបានខ្លះ ដោយ 7 Segment ប្រើអំពូល LED ចំនួន 7 គ្រាប់មកតម្រៀបគ្នាបង្កើតបានជាទម្រង់នៃលេខឡាតាំងពីលេខ 0 ដល់ ៩ ដូចរូបខាងក្រោម៖



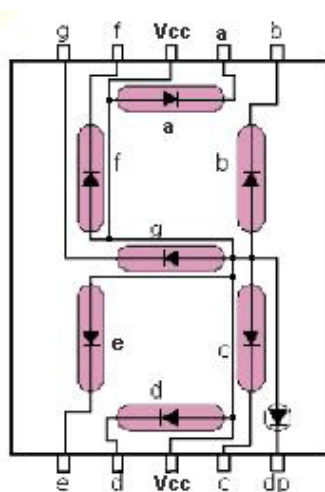
រូបទី៣.១ 7 Segment

7 Segment ត្រូវបានបែងចែកជា២ ប្រភេទគឺ៖ Common Anode និង Common cathode

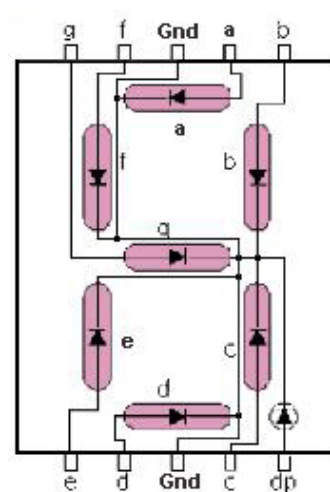
- Common Anode ជាការនាំយកជើងបូករបស់ LED ទាំង៧មករួមចូលគ្នា (common) ចំណែកជើងដែលនៅសល់ប្រើជាជើងចូល ចាំទទួលស៊ីញ៉ាល់បញ្ជាពីការសរសេរកូដរបស់យើង ហើយស៊ីញ៉ាល់បញ្ជានោះជា Logic "0" ។
- Common cathode ជាការនាំយកជើងដករបស់ LED ទាំង៧មករួមចូលគ្នា (common) ចំណែកជើងដែលនៅសល់ប្រើជាជើងចូល ចាំទទួលស៊ីញ៉ាល់បញ្ជាពីការសរសេរកូដរបស់យើង ហើយស៊ីញ៉ាល់បញ្ជានោះជា Logic "1" ។

១. 7 Segment 1digit

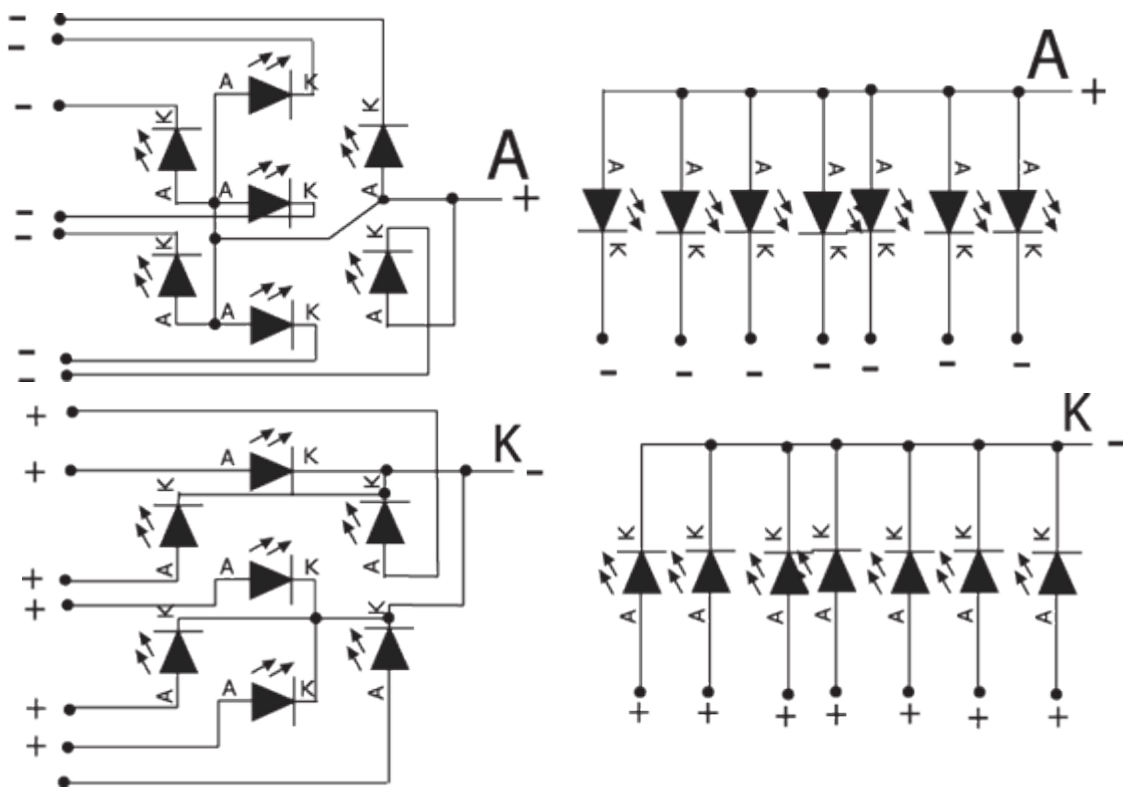
សៀវភៅខាងក្នុងរបស់ 7 Segment 1digit មានពីរប្រភេទទៅតាមប្រភេទរបស់ 7 Segment ដែរគឺ common A(អាណូដ)និង common K(កាតូដ) ។ ជាទូទៅក្នុង 7 Segment មួយមានជើងចំនួន១០ ដូចជា a, b, c, d, e, f, g, និង Vcc2 ឬ Gnd2 ដូចក្នុងរូបទី៣.២ និងរូបទី៣.៣ ។



រូបទី៣.២ Common Anode



រូបទី៣.៣ Common cathode

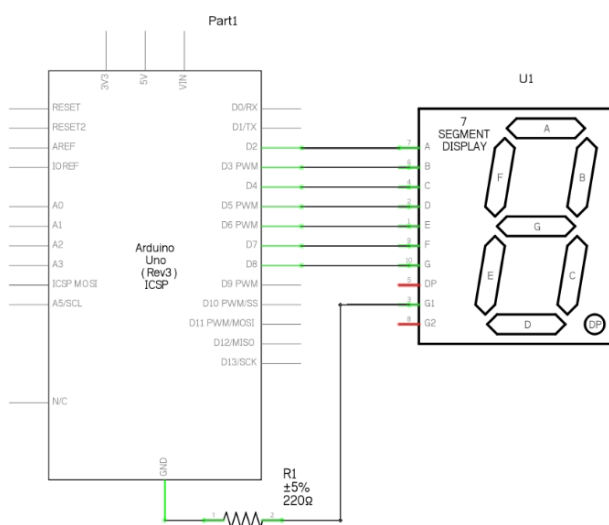


រូបទី៣.៤ សៀគ្វី 7 Segment digit 1 របស់ Common Anode និង Common cathode

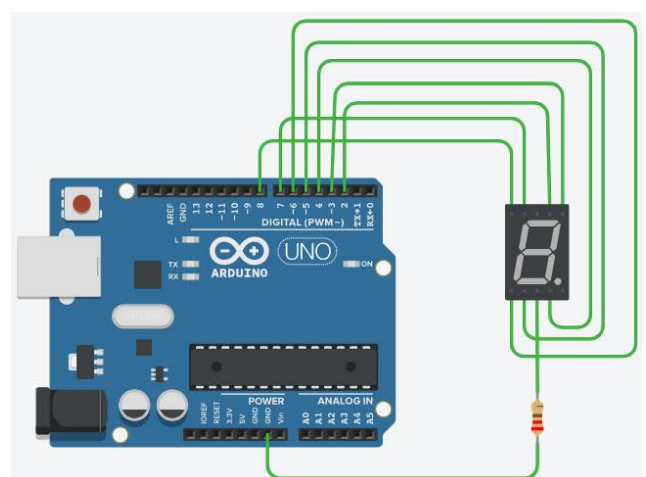
ក. ការប្រើ 7 Segment និងលេខគោល២

ជំនួសឱ្យការប្រើប្រាស់នូវ ICs ដ៏ច្រើនហើយនិងស្មុគស្មាញ Arduino board អាចកាត់បន្ថយការលំបាកនេះបានច្រើន ដោយគ្រាន់តែភ្ជាប់ជើងនីមួយៗរបស់ 7 segment ឃើងបាន Digital pins របស់ Arduino ជាការស្រេច។

ការតភ្ជាប់ 7 segment ជាមួយ board Arduino ដើម្បីសរសេរកូដបញ្ជាដូចក្នុងរូបទី៣.៥ និងរូបទី៣.៦។



រូបទី៣.៥ សៀគ្វីការតភ្ជាប់ 7 segment ជាមួយ board Arduino



រូបទី៣.៦ ការតភ្ជាប់កូដ

តាមរូបទី៣.៥ និងរូបទី៣.៦ យើងឃើញថា៖

- ជើងទី ២របស់ board Arduino តភ្ជាប់ជើង a របស់ 7 segment
- ជើងទី ៣របស់ board Arduino តភ្ជាប់ជើង b របស់ 7 segment
- ជើងទី ៤របស់ board Arduino តភ្ជាប់ជើង c របស់ 7 segment
- ជើងទី ៥របស់ board Arduino តភ្ជាប់ជើង d របស់ 7 segment
- ជើងទី ៦របស់ board Arduino តភ្ជាប់ជើង e របស់ 7 segment
- ជើងទី ៧របស់ board Arduino តភ្ជាប់ជើង f របស់ 7 segment
- ជើងទី ៨របស់ board Arduino តភ្ជាប់ជើង g របស់ 7 segment
- ជើង Gnd របស់ board Arduino តភ្ជាប់ជើង Gnd របស់ 7 segment ដោយ 7 segment

នេះជាប្រភេទ Common cathode ព្រោះជើងទី ៩និងទី ១០ របស់ 7 segment ប្រើជា Gnd។

ឧទាហរណ៍ ចូរសរសេរកូដ បញ្ជាឱ្យ 7 segment បង្ហាញលេខ ០ រហូតដល់លេខ ៧ តាមលំដាប់ ដោយតាមរូបទី៣.៥ ខាងលើ។

```
int a = 2;    // ប្រកាស a តជាមួយជើងទី២ របស់ Arduino
int b = 3;    // ប្រកាស b តជាមួយជើងទី២ របស់ Arduino
int c = 4;    // ប្រកាស c តជាមួយជើងទី២ របស់ Arduino
int d = 5;    // ប្រកាស d តជាមួយជើងទី២ របស់ Arduino
int e = 6;    // ប្រកាស e តជាមួយជើងទី២ របស់ Arduino
int f = 7;    // ប្រកាស f តជាមួយជើងទី២ របស់ Arduino
int g = 8;    // ប្រកាស g តជាមួយជើងទី២ របស់ Arduino

void setup( )
{
    pinMode(a,OUTPUT);    // កំណត់ a ជាជើងចេញ
    pinMode(b,OUTPUT);    // កំណត់ b ជាជើងចេញ
    pinMode(c,OUTPUT);    // កំណត់ c ជាជើងចេញ
    pinMode(d,OUTPUT);    // កំណត់ d ជាជើងចេញ
    pinMode(e,OUTPUT);    // កំណត់ e ជាជើងចេញ
    pinMode(f,OUTPUT);    // កំណត់ f ជាជើងចេញ
    pinMode(g,OUTPUT);    // កំណត់ g ជាជើងចេញ
}

void loop( )
{
    digitalWrite( a,1 );
    digitalWrite( b,1 );
```

```
digitalWrite( c,1 );
digitalWrite( d,1 );
digitalWrite( e,1 );
digitalWrite( f,1 );
digitalWrite( g,0 );
delay( 1000 );      // កំណត់ឱ្យចេញលេខ 0 នៅលើ 7 segment ក្នុងរយៈពេល ១វិនាទី
```

```
digitalWrite( a,0 );
digitalWrite( b,1 );
digitalWrite( c,1 );
digitalWrite( d,0 );
digitalWrite( e,0 );
digitalWrite( f,0 );
digitalWrite( g,0 );
delay( 1000 );      // កំណត់ឱ្យចេញលេខ 1 នៅលើ 7 segment ក្នុងរយៈពេល ១វិនាទី
```

```
digitalWrite( a,1 );
digitalWrite( b,1 );
digitalWrite( c,0 );
digitalWrite( d,1 );
digitalWrite( e,1 );
digitalWrite( f,0 );
digitalWrite( g,1 );
delay( 1000 );      // កំណត់ឱ្យចេញលេខ 2 នៅលើ 7 segment ក្នុងរយៈពេល ១វិនាទី
```

```
digitalWrite( a,1 );
digitalWrite( b,1 );
digitalWrite( c,1 );
digitalWrite( d,1 );
digitalWrite( e,0 );
digitalWrite( f,0 );
digitalWrite( g,1 );
delay( 1000 );      // កំណត់ឱ្យចេញលេខ 3 នៅលើ 7 segment ក្នុងរយៈពេល ១វិនាទី
```

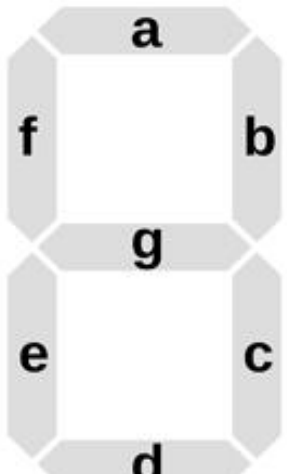
```

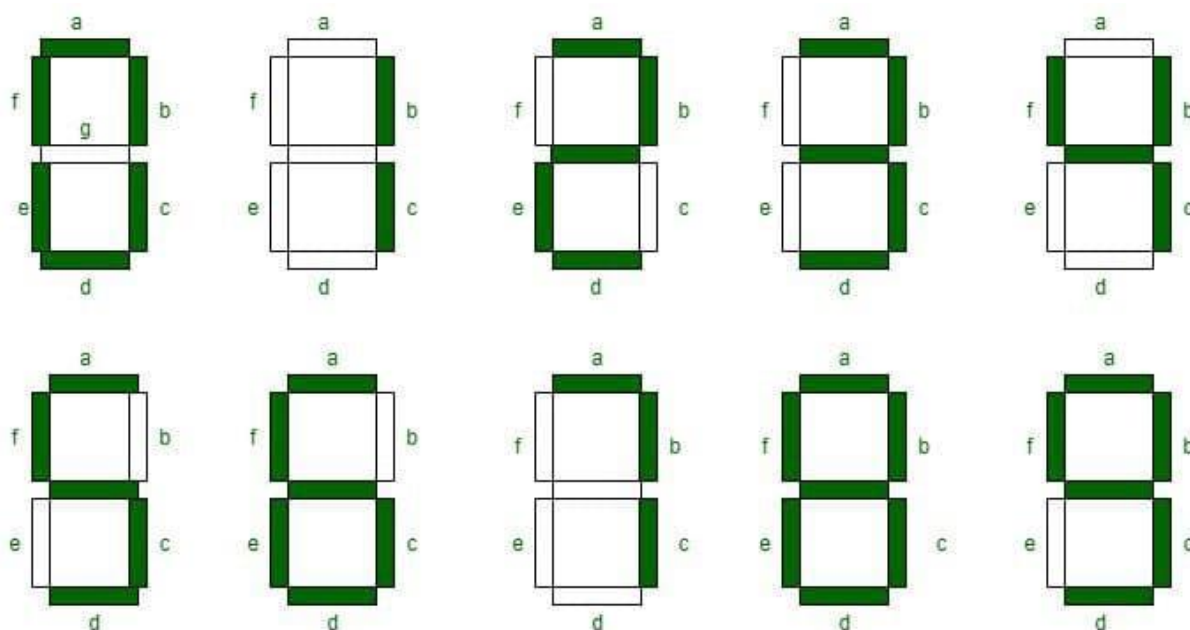
digitalWrite( a,0 );
digitalWrite( b,1 );
digitalWrite( c,1 );
digitalWrite( d,0 );
digitalWrite( e,0 );
digitalWrite( f,1 );
digitalWrite( g,1 );
delay( 1000 );      // កំណត់ឱ្យចេញលេខ 4 នៅលើ 7 segment ក្នុងរយៈពេល ១វិនាទី
digitalWrite( a,1 );
digitalWrite( b,0 );
digitalWrite( c,1 );
digitalWrite( d,1 );
digitalWrite( e,0 );
digitalWrite( f,1 );
digitalWrite( g,1 );
delay( 1000 );      // កំណត់ឱ្យចេញលេខ 5 នៅលើ 7 segment ក្នុងរយៈពេល ១វិនាទី
digitalWrite( a,1 );
digitalWrite( b,0 );
digitalWrite( c,1 );
digitalWrite( d,1 );
digitalWrite( e,1 );
digitalWrite( f,1 );
digitalWrite( g,1 );
delay( 1000 );      // កំណត់ឱ្យចេញលេខ 6 នៅលើ 7 segment ក្នុងរយៈពេល ១វិនាទី
digitalWrite( a,1 );
digitalWrite( b,1 );
digitalWrite( c,1 );
digitalWrite( d,0 );
digitalWrite( e,0 );
digitalWrite( f,0 );
digitalWrite( g,0 );
delay( 1000 );      // កំណត់ឱ្យចេញលេខ 7 នៅលើ 7 segment ក្នុងរយៈពេល ១វិនាទី
}

```

ដើម្បីឱ្យ 7 segment បង្ហាញជាលេខចេញមកបានយើងត្រូវដឹកថាគួរឱ្យ LED មួយណាភ្លឺមួយណា លេតៗ។

តារាង ប្រៀបធៀបលេខគោល១០ និងលេខគោល២

	N \ Seg	bin	g	f	e	d	c	b	a
	0	0b	0	1	1	1	1	1	1
	1	0b	0	0	0	0	1	1	0
	2	0b	1	0	1	1	0	1	1
	3	0b	1	0	0	1	1	1	1
	4	0b	1	1	0	0	1	1	0
	5	0b	1	1	0	1	1	0	1
	6	0b	1	1	1	1	1	0	1
	7	0b	0	0	0	0	1	1	1
	8	0b	1	1	1	1	1	1	1
	9	0b	1	1	0	1	1	1	1



ខ. ការប្រើ 7 Segment និងលេខគោល១៦

ដើម្បីឱ្យកូដបញ្ជា ខ្លីជាងមុនយើងអាចប្រើ លេខគោល១៦ មកជំនួសលេខគោល២ បានតាមរយៈ ការប្តូរលេខគោល២ ជាលេខគោល១៦។ ហើយត្រូវប្រើទម្រង់ថ្មីមួយទៀត គឺ៖

ទម្រង់ bitRead(n, m); ដោយ n គឺលេខដែលត្រូវអានចំនួន bit
m គឺជាលំដាប់ bit ដែលត្រូវអាន

តារាងប្រៀបធៀប លេខគោល១០និងលេខគោល២ រោង

Common Anode និង Common cathode

Common Anode								10	Common cathode							
a	b	c	d	e	f	g	HEX		HEX	a	b	c	d	e	f	g
0	0	0	0	0	0	1	0x40	0	0x3F	1	1	1	1	1	1	0
1	0	0	1	1	1	1	0x79	1	0x06	0	1	1	0	0	0	0
0	0	1	0	0	1	0	0x24	2	0x5B	1	1	0	1	1	0	1
0	0	0	0	1	1	0	0x30	3	0x4F	1	1	1	1	0	0	1
1	0	0	1	1	0	0	0x19	4	0x66	0	1	1	0	0	1	1
0	1	0	0	1	0	0	0x12	5	0x6D	1	0	1	1	0	1	1
0	1	0	0	0	0	0	0x02	6	0x7D	1	0	1	1	1	1	1
0	0	0	1	1	0	1	0x58	7	0x07	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0x00	8	0x7F	1	1	1	1	1	1	1
0	0	0	0	1	0	0	0x10	9	0x6F	1	1	1	1	0	1	1

ឧទាហរណ៍១ ចូរសរសេរកូដ បញ្ជាឱ្យ 7 segment បង្ហាញលេខ ០ រហូតដល់លេខ ៩ តាមលំដាប់ ដោយប្រើលេខគោល ១៦។

```
int K;
```

```
void setup ( ) {
```

```
    pinMode ( 2,OUTPUT );
```

```
    pinMode ( 3,OUTPUT );
```

```
    pinMode ( 4,OUTPUT );
```

```
    pinMode ( 5,OUTPUT );
```

```
    pinMode ( 6,OUTPUT );
```

```
    pinMode ( 7,OUTPUT );
```

```
    pinMode ( 8,OUTPUT );
```

```
}
```

```
void loop ( ) {
```

```
    K = 0x3F;    // កំណត់ឱ្យបង្ហាញលេខ ០ នៅលើ 7 segment
```

```
    for( int i = 2; i<9; i++ )
```

```

{
    digitalWrite (i, bitRead (K,i-2));
}
delay( 1000);
K = 0x06;    // កំណត់ឱ្យបង្ហាញលេខ 1 នៅលើ 7 segment
for( int i = 2; i<9; i++)
{
    digitalWrite (i, bitRead (K,i-2));
}
delay( 1000);
K = 0x5B;    // កំណត់ឱ្យបង្ហាញលេខ 2 នៅលើ 7 segment
for( int i = 2; i<9; i++)
{
    digitalWrite (i, bitRead (K,i-2));
}
delay( 1000);
K = 0x4F;    // កំណត់ឱ្យបង្ហាញលេខ 3 នៅលើ 7 segment
for( int i = 2; i<9; i++)
{
    digitalWrite (i, bitRead (K,i-2));
}
delay( 1000);
K = 0x66;    // កំណត់ឱ្យបង្ហាញលេខ 4 នៅលើ 7 segment
for( int i = 2; i<9; i++)
{
    digitalWrite (i, bitRead (K,i-2));
}
delay( 1000);
K = 0x6D;    // កំណត់ឱ្យបង្ហាញលេខ 5 នៅលើ 7 segment
for( int i = 2; i<9; i++)
{
    digitalWrite (i, bitRead (K,i-2));
}

```

```

delay( 1000 );
    K = 0x7D;    // កំណត់ឱ្យបង្ហាញលេខ 6 នៅលើ 7 segment
    for( int i = 2; i<9; i++ )
    {
        digitalWrite ( i, bitRead ( K,i-2 ) );
    }
    delay( 1000 );
    K = 0x07;    // កំណត់ឱ្យបង្ហាញលេខ 7 នៅលើ 7 segment
    for( int i = 2; i<9; i++ )
    {
        digitalWrite ( i, bitRead ( K,i-2 ) );
    }
    delay( 1000 );
    K = 0x7F;    // កំណត់ឱ្យបង្ហាញលេខ 8 នៅលើ 7 segment
    for( int i = 2; i<9; i++ )
    {
        digitalWrite ( i, bitRead ( K,i-2 ) );
    }
    delay( 1000 );
    K = 0x6F;    // កំណត់ឱ្យបង្ហាញលេខ 9 នៅលើ 7 segment
    for( int i = 2; i<9; i++ )
    {
        digitalWrite ( i, bitRead ( K,i-2 ) );
    }
    delay( 1000 );
}

```

យើងឃើញថាកូដសរសេរវែងនិងពិបាក ដូចនេះដើម្បីឱ្យកូដនេះខ្លីជាងមុន យើងអាចប្រើទម្រង់ថ្មីមួយទៀតគឺ Array type។

Array type ជាការប្រមូលផ្តុំនៃអថេរច្រើនចូលក្នុងកន្សោមតែមួយ ហើយត្រូវបានគេហៅប្រើតាមរយៈការហៅធាតុ ឬសន្ទសីរបស់វា។

ឧទាហរណ៍២ ចូរសរសេរកូដ បញ្ជាឱ្យ 7 segment បង្ហាញលេខ 0 រហូតដល់លេខ ៩ តាមលំដាប់ ជាមួយទម្រង់ Array type ដោយប្រើលេខគោល ២។

```

int digit1[10] = { 0b0111111, 0b0000110,
                  0b1011011, 0b1001111,
                  0b1100110, 0b1101101,
                  0b1111101, 0b0000111,
                  0b1111111, 0b1101111 };

void setup( ) {
    pinMode ( 2,OUTPUT );
    pinMode ( 3,OUTPUT );
    pinMode ( 4,OUTPUT );
    pinMode ( 5,OUTPUT );
    pinMode ( 6,OUTPUT );
    pinMode ( 7,OUTPUT );
    pinMode ( 8,OUTPUT );
}

void loop( ) {
    for( int i = 0;i<10; i++ )
    {
        for( int t = 2 ;t < 9; t++ )
        {
            digitalWrite( t,bitRead( digit1[i],t-2 ) );
            delay( 5 );
        }
        delay( 1000 );
    }
}

```

ឧទាហរណ៍៣ ចូរសរសេរកូដ បញ្ជាឱ្យ 7 segment បង្ហាញលេខ ០ រហូតដល់លេខ ៩ តាមលំដាប់ ជាមួយ ទម្រង់ Array type ដោយប្រើលេខគោល ១៦។

```

int digit1[10] = { 0x40,0x79,0x24,0x30,0x19,0x12,0x02,0x78,0x00,0x10 };
void setup( ) {
    pinMode ( 2,OUTPUT );
    pinMode ( 3,OUTPUT );
    pinMode ( 4,OUTPUT );

```

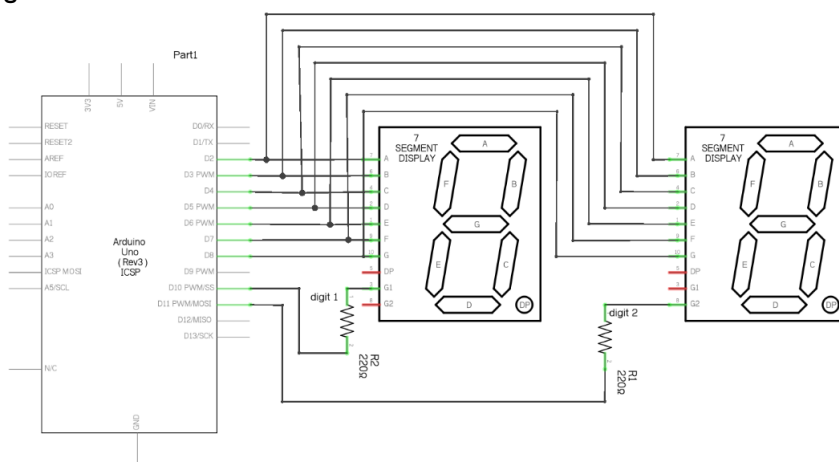
```

pinMode ( 5,OUTPUT );
pinMode ( 6,OUTPUT );
pinMode ( 7,OUTPUT );
pinMode ( 8,OUTPUT );
}
void loop( ) {
    for( int i = 0;i<10; i++)
    {
        for( int t = 2 ;t < 9; t++)
        {
            digitalWrite( t,bitRead( digit1[i],t-2 ) );
            delay( 5 );
        }
        delay( 1000 );
    }
}

```

២. 7 Segment 2digit

ចំពោះការប្រើប្រាស់ Arduino សម្រាប់លេខពីរខ្ទង់ មានភាពលំបាកជាងលេខតែមួយខ្ទង់ ការសរសេរនៅដដែលតែខុសត្រង់បច្ចេកទេសមួយចំនួនដែលយើងគួរដឹង។ សួរថាបើ 7 segments មានដឹងចំនួន ៧ ត្រូវប្រើជាចាំបាច់ អញ្ចឹងពេលប្រើពីរខ្ទង់ត្រូវការដឹង ១៤ ឬយ៉ាងណា? បើដូច្នេះមែនតើត្រូវប្តូរ Board Arduino មែនទេ? បើដឹងរបស់ Arduino uno យើងមានតែ ១៣ ដឹង? ហើយចុះបើលេខកាន់តែច្រើនខ្ទង់ទៀតតើយើងត្រូវគិតដូចម្តេច? បច្ចេកទេសដែលត្រូវបានគេប្រើសម្រាប់ករណីនេះគឺការតម្លៃទាំងប្រាំពីរនោះជាខ្ទង់ដូចខាងក្រោម៖



រូបទី៣.៧ សៀវភៅការតភ្ជាប់ 7 segment digit 2 ជាមួយ board Arduino

យើងច្បាស់ជាចូលហើយថាបើ តង់ង់ចូលគ្នាដូចនេះគឺច្បាស់ជាចេញលេខដូចគ្នា នៅលើខ្ទង់ទាំងពីរ ហើយ។ តែការពិតទៅយើងមានវិធី ដែលធ្វើឱ្យវាមិនបង្ហាញទិន្នន័យដូចគ្នាបាន តាមរយៈការធ្វើ Shift ខ្ទង់ នីមួយៗក្រោមប្រេកង់ខ្ពស់ជាងល្បឿនភ្នែករបស់យើងចាប់បាន នោះយើងនឹងមើលមិនឃើញ ថាវាដូចគ្នា នោះឡើយ។ ក្នុងឧទាហរណ៍ខាងក្រោមជាគំរូការសរសេរកូដសម្រាប់លេខពីរខ្ទង់។

ឧទាហរណ៍១ ចូរសរសេរកូដបង្ហាញលេខ 25 លើ 7 segments ពីរខ្ទង់។

```
int digit [10] = {0x40,0x79,0x24,0x30,0x19,0x12,0x02,0x78,0x00,0x10};
```

```
void setup ( ) {
```

```
    pinMode ( 2, OUTPUT );
```

```
    pinMode ( 3, OUTPUT );
```

```
    pinMode ( 4, OUTPUT );
```

```
    pinMode ( 5, OUTPUT );
```

```
    pinMode ( 6, OUTPUT );
```

```
    pinMode ( 7, OUTPUT );
```

```
    pinMode ( 8, OUTPUT );
```

```
    pinMode ( 10, OUTPUT );
```

```
    pinMode ( 11, OUTPUT );
```

```
}
```

```
void loop ( ) {
```

```
    digitalWrite( 10, HIGH );
```

```
    digitalWrite( 11, LOW );
```

```
    for ( int t = 2;t < 9;t++ ) {
```

```
        digitalWrite( t, bitRead( digit [2],t-2 ) );    // show Number 2 }
```

```
    }
```

```
    delay ( 10 );
```

```
    digitalWrite( 10, LOW );
```

```
    digitalWrite( 11, HIGH );
```

```
    for ( int t= 2;t< 9;t++ ) {
```

```
        digitalWrite ( t, bitRead( digit [5],t-2 ) );    // show Number 5 }
```

```
    }
```

```
    delay ( 10 );
```

```
}
```

ឧទាហរណ៍២ ចូរសរសេរកូដបង្ហាញលេខ ០០ ដល់ ៩៩ លើ ៧ segments ពីរខ្ទង់។

```
int digit [10] = {0x40,0x79,0x24,0x30,0x19,0x12,0x02,0x78,0x00,0x10};
```

```
void setup ( ) {
```

```
    pinMode ( 2, OUTPUT );
```

```
    pinMode ( 3, OUTPUT );
```

```
    pinMode ( 4, OUTPUT );
```

```
    pinMode ( 5, OUTPUT );
```

```
    pinMode ( 6, OUTPUT );
```

```
    pinMode ( 7, OUTPUT );
```

```
    pinMode ( 8, OUTPUT );
```

```
    pinMode ( 10, OUTPUT );
```

```
    pinMode ( 11, OUTPUT );
```

```
}
```

```
void loop ( ) {
```

```
    for( int b = 00; b <99;b++){
```

```
        int digit1 = b/10;
```

```
        int digit2 = b%10;
```

```
        for ( int i = 0;i < 50;i++) {
```

```
            digitalWrite( 10, HIGH );
```

```
            digitalWrite( 11, LOW );
```

```
            for ( int t = 2;t < 9;t++)
```

```
            {
```

```
                digitalWrite( t, bitRead( digit[digit1],t-2 ) );    // show Number 2 }
```

```
            }
```

```
            delay ( 10 );
```

```
            digitalWrite( 10, LOW );
```

```
            digitalWrite( 11, HIGH );
```

```
            for ( int t= 2;t< 9;t++)
```

```
            {
```

```
                digitalWrite ( t, bitRead( digit[digit2],t-2 ) );    // show Number 7 }
```

```
            }
```

```
            delay ( 10 );
```

```
        } } }
```

ឧទាហរណ៍ ៣ ចូរសរសេរកូដបង្ហាញលេខ 99 ដល់ 00 លើ 7 segments ពីរខ្ទង់។

```
int digit [10] = {0x40,0x79,0x24,0x30,0x19,0x12,0x02,0x78,0x00,0x10};
```

```
void setup ( ) {
```

```
    pinMode ( 2, OUTPUT );
```

```
    pinMode ( 3, OUTPUT );
```

```
    pinMode ( 4, OUTPUT );
```

```
    pinMode ( 5, OUTPUT );
```

```
    pinMode ( 6, OUTPUT );
```

```
    pinMode ( 7, OUTPUT );
```

```
    pinMode ( 8, OUTPUT );
```

```
    pinMode ( 10, OUTPUT );
```

```
    pinMode ( 11, OUTPUT );
```

```
}
```

```
void loop ( ) {
```

```
    for( int b = 99; b >=0;b--){
```

```
        int digit1 = b/10;
```

```
        int digit2 = b%10;
```

```
        for ( int i = 0;i < 50;i++ ) {
```

```
            digitalWrite( 10, HIGH );
```

```
            digitalWrite( 11, LOW );
```

```
            for ( int t = 2;t < 9;t++ )
```

```
            {
```

```
                digitalWrite( t, bitRead( digit[digit1],t-2 ) );           // show Number 2}
```

```
            }
```

```
            delay ( 10 );
```

```
            digitalWrite( 10, LOW );
```

```
            digitalWrite( 11, HIGH );
```

```
            for ( int t= 2;t< 9;t++ )
```

```
            {
```

```
                digitalWrite ( t, bitRead( digit[digit2],t-2 ) );           // show Number 7}
```

```
            }
```

```
            delay ( 10 );
```

```
        } } }
```

ក.ការបង្កើត Function declaration

សម្រាប់ការបង្កើត Function ក្នុង Arduino មានពីរប្រភេទ៖

១. Function ដែលមិន return results

ទម្រង់

```
void somename ( argument )
```

```
{
```

```
    // សរសេរកូដ
```

```
}
```

```
int digit [10] = {0x40,0x79,0x24,0x30,0x19,0x12,0x02,0x78,0x00,0x10};
```

```
void setup ( ) {
```

```
    pinMode ( 2, OUTPUT );
```

```
    pinMode ( 3, OUTPUT );
```

```
    pinMode ( 4, OUTPUT );
```

```
    pinMode ( 5, OUTPUT );
```

```
    pinMode ( 6, OUTPUT );
```

```
    pinMode ( 7, OUTPUT );
```

```
    pinMode ( 8, OUTPUT );
```

```
    pinMode ( 10, OUTPUT );
```

```
    pinMode ( 11, OUTPUT );
```

```
}
```

```
void loop ( ) {
```

```
    sendbit( 1,0,2);    // បង្ហាញលេខ 2 នៅខ្ទង់ទី១
```

```
    sendbit( 0,1,5);    // បង្ហាញលេខ 5 នៅខ្ទង់ទី២
```

```
}
```

```
void sendbit( boolean dig1, boolean dig2,int t){
```

```
    digitalWrite( 10,dig1 );
```

```
    digitalWrite( 11,dig2 );
```

```
    for ( int k=2;k<9;k++){
```

```
        digitalWrite( k,bitRead( digit[t],k-2 ) );
```

```
    }
```

```
    delay ( 10 );
```

```
}
```

២. Function Return Results

ទម្រង់ Data Type somename (argument)

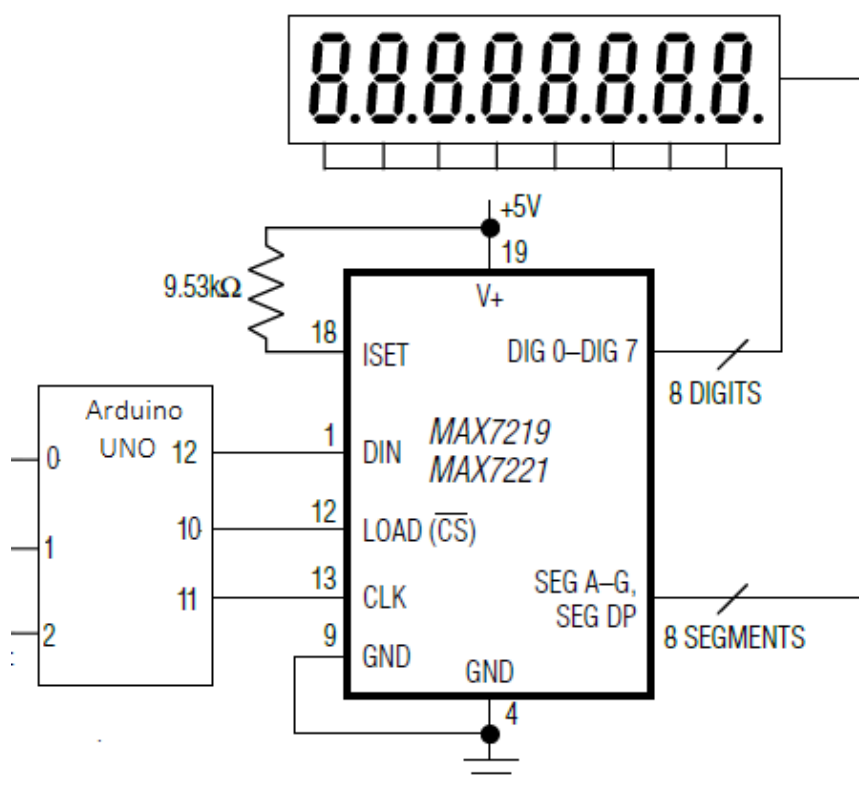
```
{
    // សរសេរកូដ
}
```

```
int x;
void setup( ) {
}
void loop( ) {
    x = calcul( 2,4,2);
}

int calcul( int i,int j,int k){
    int n;
    n = (i+j)/k;
    return n;
}
```

៣. Driver for 7 Segment 2 More than digit (Max7219)

ចំពោះវិធីសាស្ត្រខាងលើក្នុងការបង្ហាញលេខពីរខ្ទង់វាមានភាពងាយស្រួលតែក្នុងករណីដែលកូដតិច និងមិនមានកិច្ចការច្រើនសម្រាប់ឱ្យ Arduino ធ្វើការ។ តែប្រសិនបើមានកូដច្រើនយើងមិនអាចប្រើបានដូចមុនទេ ព្រោះប្រសិនបើមាន delay នៅក្នុងនោះ Shit 7 segment របស់យើងនឹងយឺតហើយ យើងច្បាស់ជាអាចមើលវាទាន់។ ដូច្នេះការប្រើប្រាស់ IC Driver សម្រាប់ករណីនេះស្រួលចំណេញទាំងពេលវេលា និង ទាំង Hardware ។ ទោះយ៉ាងណាក៏ នៅមានការលំបាកដែរសម្រាប់ អ្នកទើបចាប់ផ្តើមប្រើដំបូងនិងមាន Protocol ស្មុគស្មាញបន្តិច។ ខាងក្រោមជាសៀវភៅសម្រាប់ត 7 segment ជាមួយ IC Max7219។

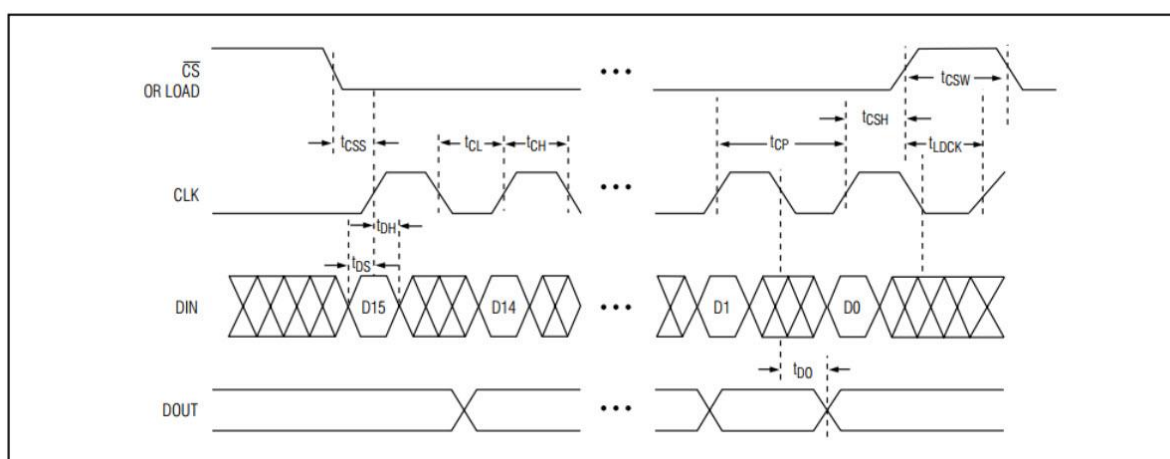


រូបទី៣.៨ សៀគ្វីការត IC 7219 ជាមួយ 7 segment ប្រើនិ digit

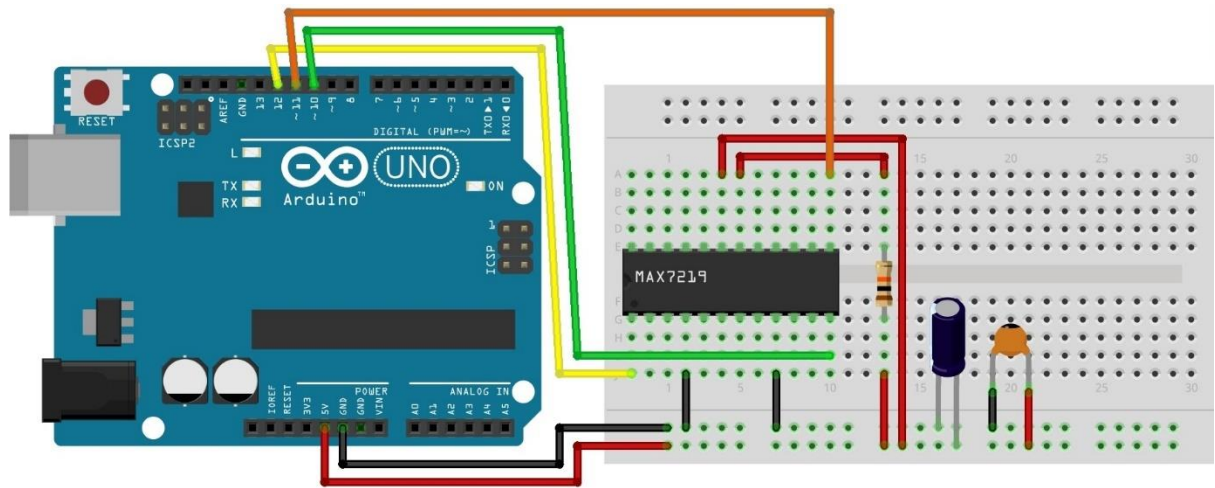
តាមតារាងបង្ហាញថា Max7219 ធ្វើការទទួលទិន្នន័យម្តងចំនួន 16 bit ដោយពី bit ទី ០ - ៧ ជាលេខត្រូវបង្ហាញនៅ address ហើយ ចាប់ពី bit ទី ៨ - ១១ ជា address គឺសំដៅលើទីតាំងរបស់ខ្ទង់ណាមួយត្រូវបង្ហាញលេខ និងចាប់ពី bit ទី ១២ - ១៥ មិនគិត។

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
x	x	x	x	ADDRESS				M58				DATA			L58

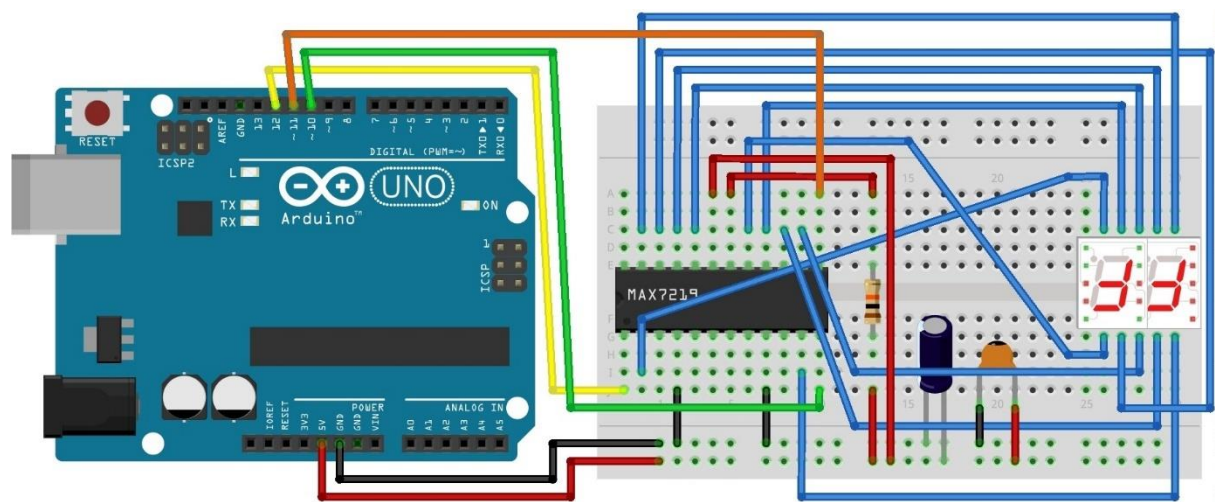
ហើយចំពោះការបញ្ចូលទិន្នន័យនៃ bit នីមួយៗគឺត្រូវយកតាមតារាង Time Diagram ខាងក្រោម bit ទី ១៥ ត្រូវបញ្ជូនទៅមុនគេហើយ bit បន្ទាប់ៗទៀតត្រូវបញ្ចូលរៀងរាល់ clk Rising edge និង load ស្មើម្តងមួយពេលគ្រប់ 16 bit ។



រូបទី៣.៩ សៀគ្វីការធ្វើការរបស់ IC Max 7219



រូបទី៣.១០ សៀគ្វីការត IC 7219 ជាមួយ board Arduino



រូបទី៣.១១ សៀគ្វីការត IC 7219 ជាមួយ board Arduino នឹង 7 segment digit 2

គំរូកូដសម្រាប់ប្រតិបត្តិការបញ្ជូនទិន្នន័យចេញពី Arduino ទៅកាន់ IC Max7219

```
const int load = 10;
```

```
const int clk =11;
```

```
const int din = 12;
```

```
void setup ( ) {
```

```
    pinMode ( din, OUTPUT);
```

```
    pinMode ( load, OUTPUT);
```

```
    pinMode ( clk, OUTPUT);
```

```
}
```

```
void loop ( ) {
```

```
}
```

```

void sda( byte add,byte da ){
    digitalWrite ( load,LOW );
    for ( int i = 7; i >= 0; i -- ) {
        digitalWrite( din, bitRead( add,i ) );
        digitalWrite( clk, LOW );
        digitalWrite( clk, HIGH );
    }
    for( int i = 7;i >= 0;i--){
        digitalWrite( din, bitRead ( da,i ) );
        digitalWrite( clk, LOW );
        digitalWrite( clk, HIGH );
    }
    digitalWrite( load, HIGH );
}

```

ទម្រង់

Const data type constant name = value;

ដោយ constant name : គឺឈ្មោះណាមួយ

Value : គឺតម្លៃរបស់វា

ឧទាហរណ៍

```
const float pi = 3.14;
```

```
const float gravity = 9.8;
```

```
const int clk = 9;
```

ខាងលើគ្រាន់តែជាប្រតិបត្តិការនៃការបញ្ជូនទិន្នន័យទៅកាន់ Max7219 តែវាមិនទាន់គ្រប់គ្រាន់នៅឡើយទេ យើងត្រូវ register នៅ address មួយចំនួនទៀតទើបអាចប្រើប្រាស់បាន។

តារាង address to register

REGISTER	ADDRESS					HEX CODE
	D15-D12	D11	D10	D9	D8	

No-Op	X	0	0	0	0	0xXo
Digit 0	X	0	0	0	1	0xXo1
Digit 1	X	0	0	1	0	0xXo2
Digit 2	X	0	0	1	1	0xXo3
Digit 3	X	0	1	0	0	0xXo4
Digit 4	X	0	1	0	1	0xXo5
Digit 5	X	0	1	1	0	0xXo6
Digit 6	X	0	1	1	1	0xXo7
Digit 7	X	1	0	0	0	0xXo8
Decode Mode	X	1	0	0	1	0xXo9
Intensity	x	1	0	1	0	0xXoA
Scan Limit	X	1	0	1	1	0xXoB
Shutdown	X	1	1	0	0	0xXoC
Display Test	X	1	1	1	1	0xXoF

Address នីមួយៗត្រូវភ្ជាប់ជាមួយ data ដើម្បីបានជាលទ្ធផលណាមួយដែលយើងចង់បាន ការដាក់តម្លៃយ៉ាងណានោះត្រូវមើលតាម datasheet របស់វា។

ក. Shutdown Mode (0xXC);

ប្រើសម្រាប់ធ្វើការបិទបើកប្រតិបត្តិការរបស់ Max7219។ នៅពេលដែល data bit ទី០ស្មើ១ នោះ IC Max7219 ស្ថិតក្នុង Normal operation ហើយផ្ទុយមកវិញបើ data bit ទី០ស្មើ ០ គឺមានន័យថាស្ថិតក្នុងម៉ូតដែលមិនឱ្យដំណើរការ។

MOOE	ADDRESS COOE (HEX)	REGISTER DATA							
		D7	D6	D5	D4	D3	D2	D1	D0
Shut down Mode	0xXC	x	x	x	x	x	x	x	0
Normal Operation	0xXC	x	x	x	x	x	x	x	1

២. Decode Mode (0xX9);

ជាម៉ូតសម្រាប់កំណត់ឱ្យ IC Max 7219 បំប្លែងលេខ (decode binary number) ដែលយើងបញ្ជូនទៅ ឱ្យទៅជាលេខសម្រាប់ 7 segment ឬ អត់។

DECOOE MOOE	REGISTER DATA								HEX COOE
	D7	D6	D5	D4	D3	D2	D1	D0	
No decode for digits 7-0	0	0	0	0	0	0	0	1	0x00
Code B decode for digits 0 No decode for digits 7-1	0	0	0	0	0	0	0	1	0x01
Code B decode for digits 3-0 No decode for digits 7-4	0	0	0	0	1	1	1	1	0x0F
Code B decode for digits 7-0	1	1	1	1	1	1	1	1	0xFF

ម៉ូតនេះស្រដៀងគ្នានឹង IC74is47 ដែលយើងធ្លាប់រៀនពីមុនដែរគ្រាន់តែថាវាប្រើសម្រាប់ 7 segment cc ។ ខាងក្រោមជាតារាងលេខរបស់វាដែលអាចបង្ហាញលើអង្កត់លេខ។

7-SEOMENT CHATACTER	REGISTER DATA						ON SEGMENTS = 1							
	D7	D6- D4	D3	D2	D1	D0	DP	A	B	C	D	E	F	G
0		X	0	0	0	0		1	1	1	1	1	1	0
1		X	0	0	0	1		0	1	1	0	0	0	0
2		X	0	0	1	0		1	1	0	1	1	0	1
3		X	0	0	1	1		1	1	1	1	0	0	1
4		X	0	1	0	0		0	1	1	0	0	1	1
5		X	0	1	0	1		1	0	1	2	0	1	1
6		X	0	1	1	0		1	0	1	1	1	1	1
7		X	0	1	1	1		1	1	1	0	0	0	0
8		X	1	0	0	0		1	1	1	1	1	1	1

9		X	1	0	0	1		1	1	1	1	0	1	1
—		X	1	0	1	0		0	0	0	0	0	0	1
E		X	1	0	1	1		1	0	0	1	1	0	1
H		X	1	1	0	0		0	1	1	0	1	1	1
L		X	1	1	0	1		0	0	0	1	1	1	0
P		X	1	1	1	0		1	1	0	0	1	1	1
blank		X	1	1	1	1		0	0	0	0	0	0	0

គ. Scan Limit (0xB);

ប្រើសម្រាប់កំណត់ចំនួនខ្ទង់ដែលត្រូវបង្ហាញតាមតម្រូវការរបស់យើងបាន ៖

SCAN LIMIT	REGISTER DATA								HEX
	D7	D6	D5	D4	D3	D2	D1	D0	COOE
Display digit 0 only	X	X	X	X	X	0	0	0	0x0
Display digits 0 0 &1	X	X	X	X	X	0	0	1	0x1
Display digits 0 1 2	X	X	X	X	X	0	1	0	0x2
Display digits 0 1 2 3	X	X	X	X	X	0	1	1	0x3
Display digits 0 1 2 3 4	X	X	X	X	X	1	0	0	0x4
Display digits 0 1 2 3 4 5	X	X	X	X	X	1	0	1	0x5
Display digits 0 1 2 3 4 5 6	X	X	X	X	X	1	1	0	0x6
Display digits 0 1 2 3 4 5 6 7	X	X	X	X	X	1	1	1	0x7

ង. Intensity (0xA);

ប្រើសម្រាប់កំណត់កម្រិតពន្លឺរបស់ 7 segment តាមរយៈ Programmable ដែលមានន័យថា យើងអាចធ្វើឱ្យពន្លឺខ្លាំងឬខ្សោយតាមរយៈការ register ទៅកាន់ address នេះនិង add ណាមួយដែលយើងត្រូវការ ៖

DUTY CYCLE		D7	D6	D5	D4	D3	D2	D1	D0	HEX COOE
MAX7219	MAX 7221									
1/6 (min on)		x	X	X	X	0	0	0	0	0xXo
2/16		X	X	X	X	0	0	0	1	0xX1
3/16		X	X	X	X	0	0	1	0	0xX2
4/16		X	X	X	X	0	0	1	1	0xX3
5/16		X	X	X	X	0	1	0	0	0xX4
6/16		X	X	X	X	0	1	0	1	0xX5
7/16		X	X	X	X	0	1	1	0	0xX6
8/16		X	X	X	X	0	1	1	1	0xX7
9/16		X	X	X	X	0	0	0	0	0xX8
10/16		X	X	X	X	1	0	0	1	0xX9
11/16		X	X	X	X	1	0	1	0	0xXA
12/16		X	X	X	X	1	0	1	1	0xXB
13/16		X	X	X	X	1	1	0	0	0xXC
14/16		X	X	X	X	1	1	0	1	0xXD
15/16		X	X	X	X	1	1	1	0	0xXE
15/16 (max on)		X	X	X	X	1	1	1	1	0xFF

៥. Display Test (0xFF);

ប្រើសម្រាប់ធ្វើតេស្តចង់ដឹងថាមានអង្កត់ LED ណាមួយមិនភ្លឺឬអត់ តែត្រូវចាំថានៅពេលដែលយើងធ្វើតេស្តរួច យើងត្រូវបិទ register នេះវិញដើម្បីត្រលប់មកភាពដើម្បី៖

MODE	REGISTER DATA							
	D7	D6	D5	D4	D3	D2	D1	D0
Nomal Operation	X	X	X	X	X	X	X	0
Display Test Mode	X	X	X	X	X	X	X	1

ឧទាហរណ៍ សរសេរកូដឱ្យ IC Max 7219 បង្ហាញលេខ 37

```

const int din = 8;
const int load = 10;
const int clk = 9;
void setup ( ) {
    pinMode ( din,OUTPUT );
    pinMode ( load,OUTPUT );
    pinMode ( clk,OUTPUT );
    sda ( 0x0c,0x01 );    // Register to Nomal Mode
    sda ( 0x09,0xff );    // Register Decode mode
    sda ( 0x0a,0x07 );    // Register Intensity
    sda ( 0x0b,0x07 );    // Register Scan limited
}
void loop ( ) {
    sda ( 2,3 );
    sda ( 1,7 );
}
void sda ( byte add, byte da ) {
    digitalWrite ( load, LOW );
    for ( int i = 7; i>=0; i-- ) {
        digitalWrite( din, bitRead ( add, i ) );
        digitalWrite( clk, LOW );
        digitalWrite( clk, HIGH );
    }
    for ( int i= 7; i>=0; i-- ) {
        digitalWrite( din, bitRead ( add, i ) );
        digitalWrite( clk, LOW );
        digitalWrite( clk, HIGH );
    }
    digitalWrite( load, HIGH ); }

```

ជាមួយគ្នានេះដែរ Arduino IDE ក៏មាន ទម្រង់ថ្មីមួយទៀតអាចប្រើប្រាស់បានជាមួយការបញ្ចូល ទិន្នន័យក្នុង Protocol នេះទៅកាន់ IC Max 7219 បានដែលមានឈ្មោះថា shiftOut ។

ទម្រង់ ShiftOut

ShiftOut (data Pin, Clock Pin, bit Order, Value)

ដោយ data Pin: លេខជើងដែលភ្ជាប់ទៅកាន់ DIN របស់ IC Max7219

Clock Pin: លេខជើងដែលភ្ជាប់ទៅកាន់ CLK របស់ IC Max7219

Bit Order: មានពីរ MSBFIRST OR LSBFIRST

Value: តម្លៃមួយ byte ដែលត្រូវចេញ

ឧទាហរណ៍

```
#define din 7
#define load 5
#define clk 6

void setup ( ) {
    pinMode ( din, OUTPUT );
    pinMode ( load, OUTPUT );
    pinMode ( clk, OUTPUT );
    sda( 0x0c,0x01 );
    sda( 0x09,0xff );
    sda( 0x0a,0x07 );
    sda( 0x0b,0x07 );
}

void loop ( ) {
    for ( int x = 1; x < 99; x++ ) {
        int dig1 = x/10;
        int dig2 = x%10;
        sda( 2,dig1 );
        sda( 1,dig2 );
        delay ( 1000 );
    }
}
```

```

    }
}

void sda( byte add, byte da ) {
    digitalWrite ( load, LOW );
    shiftOut ( din clk, MSBFIRST, add );
    shiftOut ( din clk MSBFIRST, da );
    digitalWrite ( load, HIGH );
}

```

ទម្រង់ #define Constant name vale ប្រើសម្រាប់ការប្រកាសនិមិត្តសញ្ញាតំណាងឱ្យលេខណាមួយ ឬកូដណាមួយ។

ឧទាហរណ៍

```

#define mydelay delay ( 10 );
#define door open digitalWrite ( 5, HIGH );

```

មេរៀនទី៤ Digital Input

១. Comparison Operator

ការសរសេរកម្មវិធី ដែលមានលក្ខខណ្ឌ គឺត្រូវប្រើនិមិត្តសញ្ញាក្នុងការដំណើរការឬ ប្រៀបធៀប ថាពិត ឬមិនពិត ដោយមាននិមិត្តសញ្ញាដូចខាងក្រោម៖

ប្រភេទ Competition Operator	របៀបប្រើ
$x == y$	មានន័យថា x មានតម្លៃស្មើ y
$x != y$	មានន័យថា x មានតម្លៃមិនស្មើ y
$x < y$	មានន័យថា x មានតម្លៃតូចជាង y
$x > y$	មានន័យថា x មានតម្លៃធំជាង y
$x <= y$	មានន័យថា x មានតម្លៃតូចជាងឬស្មើ y
$x >= y$	មានន័យថា x មានតម្លៃធំជាងឬស្មើ y

២. Bitwise Operator

ជាអនុគមន៍តក្កវិទ្យាដែលរួមមាន AND,OR,NOT,XOR,SHIFTLEFT,SHIFTRIGHT...។
ប្រតិបត្តិការទាំងនោះធ្វើឡើងតែនៅលើតម្លៃ bit ប៉ុណ្ណោះ ប្រតិបត្តិការទាំងនេះត្រូវបានគេហៅថា Logic Gate ។ ខាងក្រោមជាតារាងតក្ក៖

$$A \text{ AND } B = A \cdot B$$

A	B	A·B
0	0	0
0	1	0
1	0	0
1	1	1

$$A \text{ OR } B = A + B$$

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

$$\text{NOT}(A) = A$$

A	A
0	1
1	0

Arduino symbol	
AND	&
OR	
NOT	
XOR	^
Shift left	<<
Shift Right	>>

៣. Boolean Operator

ត្រូវបានប្រើសម្រាប់តេស្តមើលលក្ខខណ្ឌពីរ ឬ ច្រើន។ វាមានលក្ខណៈខុសគ្នាពី Boolean Operator ព្រោះវាជាការប្រៀបធៀបលក្ខខណ្ឌមិនមែនជាការគណនាតក្កវិទ្យានោះឡើយ។

Arduino symbol		មុខងារ
AND	&&	ប្រៀបធៀបលក្ខខណ្ឌពីរ ឬច្រើនបើដូចគ្នាទាំងអស់គឺពិត ផ្ទុយទៅវិញបើមានតែមួយខុសគឺ មិនពិត
OR		ប្រៀបធៀបលក្ខខណ្ឌពីរ ឬច្រើនបើមានមួយណាពិត គឺពិតទាំងអស់ ផ្ទុយទៅវិញបើខុសទាំងអស់ គឺមិនពិត
NOT	!	វាពិតនៅពេលតម្លៃលេខវាផ្ទុយពីតម្លៃបច្ចុប្បន្ន

៤. ការគណនាបែបគណិតវិទ្យា

ក្នុងការគណនាបែបគណិតវិទ្យាមានដូចជា បូក ដក គុណ ចែក ជាដើមដែលជាការគណនារវាងតម្លៃ 2 ផ្សេងគ្នា

$y = y + 3;$ $x = x - 7;$

$i = i * 6;$ $r = r / 5;$

ក្នុងដំណើរការគណិតវិទ្យា យើងត្រូវគិតដល់ការប្រកាសអថេរ ដូចជា ប្រសិនបើផលបូក ដក គុណ ឬចែក ជាចំនួនពេញ ត្រូវប្រកាសអថេរ int តែប្រសិនបើជាលេខក្បៀស ត្រូវប្រកាសអថេរ float ជាដើម។ ប្រមាណវិធីគណិតវិទ្យាដែលត្រូវបានគេបង្កើតឡើងដើម្បីសរសេរឱ្យបានលឿន ចំណេញពេល និងឆាប់រហ័ស ចំពោះប្រមាណវិធីបណ្តាក់គ្នាលើអថេរតែមួយ។

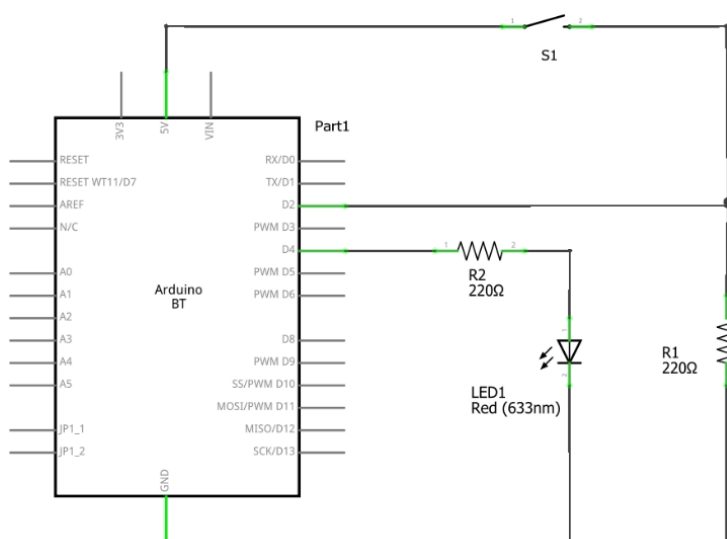
ការប្រើនិមិត្តសញ្ញា គណិតវិទ្យា ក្នុងការសរសេរកម្មវិធី

++	X++	មានន័យថា $x = x + 1$, ឬ តម្លៃ x បូកថែមដោយ 1
	++X	មានន័យថា $x = x + 1$, ឬ តម្លៃ x បូកថែមដោយ 1 តែយកតម្លៃថ្មីក្រោយបូក ១
--	X--	មានន័យថា $x = x - 1$, ឬ តម្លៃ x ដកចេញដោយ 1
	--X	មានន័យថា $x = x - 1$, ឬ តម្លៃ x ដកចេញដោយ 1 តែយកតម្លៃថ្មីក្រោយដក ១
+=	X+=Y;	មានន័យថា $x = x + y$, ឬ តម្លៃ x បូកដោយ y
-=	X-=Y;	មានន័យថា $x = x - y$, ឬ តម្លៃ x ដកដោយ y
*=	X *= Y;	មានន័យថា $x = x * y$, ឬ តម្លៃ x គុណដោយ y
/=	X /= Y;	មានន័យថា $x = x / y$, ឬ តម្លៃ x ចែកដោយ y
%=	X %= Y;	$X = x \% y$;
&=	X& = Y;	$X = x \& y$;
=	X = Y;	$X = x y$;

៥. Digital Input Arduino

របៀបតសៀវគ្គីងាយមួយបង្ហាញពីការប្រើប្រាស់ input ចូលទៅកាន់ Arduino តាមរយៈជើង Digital របស់ board Arduino ដោយតតាមរូបទី៤.១ខាងក្រោម។

ឧទាហរណ៍១ចុចភ្លឺ អត់ចុចលត់។



រូបទី៤.១ ការតសៀវគ្គី Arduino

```
void setup ( ) {
    pinMode ( 2, INPUT );
    pinMode ( 4, OUTPUT );
}

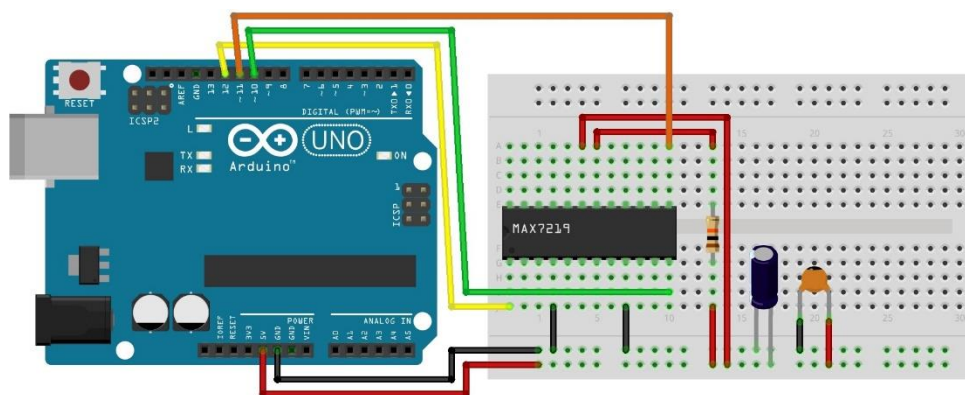
void loop ( ) {
    if ( digitalRead ( 2 ) == 1 ) {
        digitalWrite ( 4,HIGH );
    } else {
        digitalWrite ( 4,LOW );
    }
}
```

ឧទាហរណ៍២ ចូរសរសេរកូដប្រើប៊ូតុងចុចម្តងក្តី ចុចម្តងរលត់។

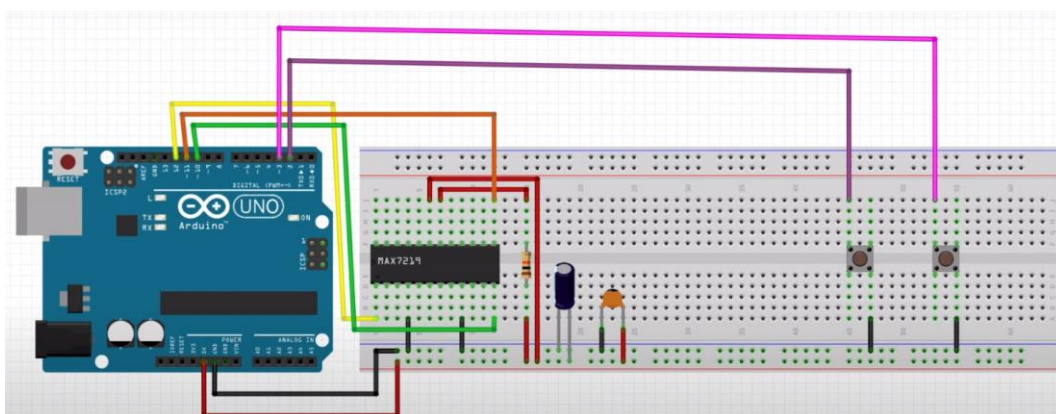
```

int sw =2;
int LED =4;
void setup ( ) {
    pinMode ( sw, INPUT );
    pinMode ( LED, OUTPUT );
}
void loop ( ) {
    if ( digitalRead ( sw ) == 1 )
    {
        digitalWrite ( LED,digitalRead( sw ) );
        delay( 100 );
    }
}

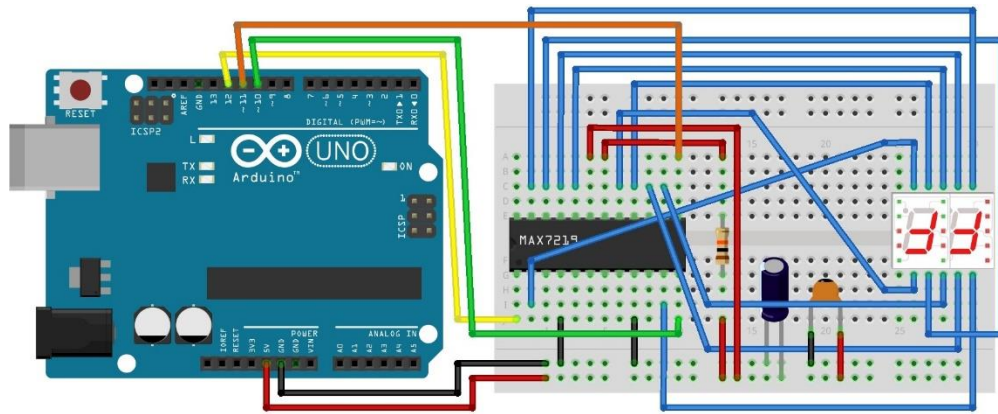
```

ឧទាហរណ៍៣ ចូរសរសេរកូដប្រើប៊ូតុងចំនួនពីរសម្រាប់រាប់លេខចុះ និងឡើងដោយប្រើ Max7219

រូបទី៤.២ ការភ្ជាប់ Max 7219 ជាមួយ board Arduino uno



រូបទី៤.៣ ការភ្ជាប់ Max 7219 ជាមួយ board Arduino uno និង កុងតាក់



រូបទី៤.៤ ការភ្ជាប់ Max 7219 ជាមួយ board Arduino uno និង 7 segment digit 2

```
#define load 10
#define clk 11
#define din 12
int up = 2;
int down = 3;
int count;
void setup( ) {
    pinMode( din,OUTPUT );
    pinMode( load,OUTPUT );
    pinMode( clk,OUTPUT );
    pinMode( up,OUTPUT );
    pinMode( down,OUTPUT );
    sda( 0x0c,0x01 );
    sda( 0x09,0xff );
    sda( 0x0a,0x02 );
    sda( 0x0b,0x01 );
}
void loop( ) {
    if ( digitalRead( up )==1 )
    {
        count++;
        delay( 300 );
    }
    if ( digitalRead( down )==1 )
```

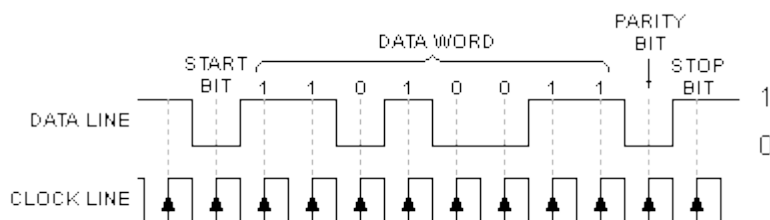
```

        {
            count--;
            delay( 300 );
        }
    if ( count < 0 )
    {
        count=0;
    }
    if ( count > 99 )
    {
        count=99;
    }
    int dig1 =count/10;
    int dig2 =count%10;
        sda( 1,dig1 );
        sda( 2,dig2 );
    }
void sda( byte add,byte da ){
    digitalWrite( load,LOW );
    shiftOut( din,clk,MSBFIRST,add );
    shiftOut( din,clk,MSBFIRST,da );
    digitalWrite( load,HIGH );
}

```

មេរៀនទី៥ Serial Monitoring និង String Function

Serial Monitor ជារបៀបមួយដែល កញ្ចប់ទិន្នន័យត្រូវបានបោះចេញម្តងមួយៗក្នុងរយៈពេលថេរ។ រយៈពេលថេរ ត្រូវបានហៅថា Baudrate ។ ស្តង់ដារ Baudrate មានដូចជា 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, or 115200។ រូបខាងក្រោមបង្ហាញពីកញ្ចប់ទិន្នន័យដែល Protocol serial ។



១. Serial of Arduino

Arduino IDE ដែលយើងប្រើមានភ្ជាប់មកជាមួយនូវ Serial Monitor ដែលអាចឱ្យយើងធ្វើតេស្តជាមួយ Serial code របស់វា ក៏ Debug ជាមួយឧបករណ៍ផ្សេងៗបាន។ ដើម្បីអាចប្រើវាបានយើងត្រូវយល់ដឹងទម្រង់មួយចំនួនដូចតទៅនេះ។

- **Begin ();**

ប្រើសម្រាប់ឱ្យ Serial port របស់ Arduino ដំណើរការជាមួយ Baudrate ណាមួយ

ទម្រង់ Serial.begin (Baudrate);

ឧទាហរណ៍

```
void setup ( )
```

```
{
```

```
    Serial.begin( 115200 );    // Initialize serial with baudrate 115200
```

```
}
```

- **if (Serial);**

ប្រើសម្រាប់ធ្វើតេស្តមើលថាតើ Serial port ត្រូវបានភ្ជាប់ហើយឬនៅ

ឧទាហរណ៍

```
void setup ( ) {
```

```
    Serial.begin( 9600 ); // Initialize serial and wait for port to open;
```

```
    While( !Serial ) {
```

```
        // wait for serial port to connect. Needed for native USB
```

```
    } }
```

- **available();**

ប្រើសម្រាប់តេស្តមើលថា តើមានទិន្នន័យនៅក្នុង Buffer មុននឹងអាននូវទិន្នន័យនោះ

ឧទាហរណ៍

```
void setup ( ) {
    Serial.begin( 9600 );
}

void loop( ) {
    if( Serial.available( )>0 ){
        int x = Serial.read( );
    }
}
```

- **end();**

ប្រើសម្រាប់បិទប្រតិបត្តិការណ៍របស់ជើង Serial ហើយឱ្យវាក្លាយទៅជាជើង Digital In/ Out ធម្មតា

ឧទាហរណ៍

```
void setup ( ) {
    Serial.end( );
    pinMode( 0,INPUT );
    pinMode( 1,INPUT );
}
```

- **find();**

ប្រើសម្រាប់រកពាក្យ (string) ណាមួយក្នុង Buffer។ ប្រសិនបើរកឃើញពាក្យនោះវានឹង return true ផ្ទុយទៅវិញ false នៅដល់ពេល timeout ។

ទម្រង់ **Serial.find(targetString);**

ឧទាហរណ៍

```
void setup ( ) {
    Serial.begin( 9600 );
}

void loop( ){
    if( Serial.available( )>0 ){
        if( Serial.find( "OK" ) ){
            int x = Serial.read( );
        }
    }
}
```

```
    }
}
```

- **print();**

ប្រើសម្រាប់បញ្ជូនទិន្នន័យពី Arduino Serial របស់វាទៅខាងក្រៅ ឬទៅឧបករណ៍ដទៃទៀត

```
Serial.print(Val);
Serial.print(Val,format);
```

ឧទាហរណ៍

```
Serial.print( 78 );           // gives 78
Serial.print( 1,23456 );     // gives 78
Serial.print( "N" );         // gives N
Serial.print( "Hello KCIT" ); // gives Hello KCIT
Serial.print( 78, BIN );     // gives print ចេញលេខគោល២ "1001110"
Serial.print( 78, DEC );     // gives print ចេញលេខគោល១០ "78"
Serial.print( 78, HEX );     // gives print ចេញលេខគោល១៦
Serial.println( 1.23456,0 ); // gives "1"
Serial.println( 1.23456,2 ); // gives "1.23"
Serial.println( 1.23456,4 ); // gives "1.2345"
```

- **println();**

ប្រើសម្រាប់បញ្ជូនទិន្នន័យពី Arduino Serial ទៅខាងក្រៅ ឬឧបករណ៍ដទៃតែមានភ្ជាប់ជាមួយ ASCII (ASCII 10 or "\n") ចុះបន្ទាត់។

ឧទាហរណ៍

```
int count;
void setup ( ) {
    Serial.begin( 9600 );
}
void loop( ){
    count++;
    Serial.println( count );
}
```

- **read();**

សម្រាប់ទាញយកទិន្នន័យ 1byte ពីក្នុង Buffer ហើយទិន្នន័យដែលទទួលបានគឺជាតម្លៃលេខ។

ឧទាហរណ៍

```
void setup ( ) {
    Serial.begin( 9600 );
}
void loop( )
{
    if( Serial.available( )>0 ); {
        int x = Serial.read( );
    }
}
```

- **readString();**

សម្រាប់អានគ្រប់ទិន្នន័យទាំងអស់ក្នុង Buffer រួចបំប្លែងជាពាក្យចេញមក

ទម្រង់ **Serial.readString();**

ឧទាហរណ៍

```
void setup ( ) {
    Serial.begin( 9600 );
}
void loop( )
{
    if( Serial.available( )>0 ); {
        txt = Serial.readString( );
    }
}
```

- **setTimeout ();**

ប្រើសម្រាប់កំណត់រយៈពេល Arduino ក្នុងការធ្វើការអានទិន្នន័យ ប្រសិនបើមិនអានក្នុងរយៈពេលកំណត់ឱ្យអស់ទេ នោះវានឹងបាត់បង់ទិន្នន័យខ្លះ។

ទម្រង់ **Serial.setTimeout(Duration);**

ឧទាហរណ៍

```

String txt;

void setup ( ) {
    Serial.begin( 9600 );
    Serial.setTimeout( 100 );
}

void loop( ){
    if( Serial.available( )>0 ){
        txt = Serial.readString( );
    }
}

```

- **write ();**

បញ្ជូនទិន្នន័យ 1byte ចេញពី Arduino ទៅខាងក្រៅ

ឧទាហរណ៍

```

byte x = 90;

void setup ( ) {
    Serial.begin( 9600 );
}

void loop( ){
    Serial.write(x);
}

```

អនុវត្ត

ចូរសរសេរកូដបញ្ជាក់ក្នុងតាក់បុចបុះ ចុចឡើង ពីលេខ ០០ ដល់ ៩៩ ឬពី ៩៩ ដល់ ០០ ដោយឱ្យបង្ហាញលេខនៅលើ Serial monitor។

```

int up = 2;
int down = 3;
int count;

void setup ( ) {
    pinMode(up,OUTPUT);
}

```

```

        pinMode( down,OUTPUT );
        Serial.begin( 9600 );
    }
    void loop( ){
        if ( digitalRead( up )==1 )
        {
            count++;
            delay( 300 );
        }
        if ( digitalRead( down )==1 )
        {
            count--;
            delay( 300 );
        }
        if ( count < 0 )
        {
            count=0;
        }
        if ( count > 99 )
        {
            count=99;
        }
        Serial.println( count );
        delay( 100 );
    }

```

២. String Objects

String ជាបណ្តុំនៃអក្សរច្រើនចូលគ្នាបង្កើតបានជាពាក្យ។ អក្សរនីមួយៗមានតម្លៃស្មើ 1byte ហើយគោរពទៅតាមតារាង ASCII ។

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

នៅក្នុង Arduino បានផ្តល់ឱ្យនូវ Function មួយចំនួនសម្រាប់ប្រើប្រាស់ជាមួយ String នេះមានដូចជា

- String ()
- charAT ()
- compareTo ()
- concat ()
- endsWith ()
- equals ()
- equalsIgnoreCase ()
- indexOf ()
- lastIndexOf ()
- length ()
- remove ()
- replace ()
- setCharAt ()
- substring ()
- toInt ()
- toInt ()
- toFloat ()
- toLowerCase ()
- toUpperCase ()
- trim ()

- String (); បង្កើតចេញពី Data Type ផ្សេងៗ។

ទម្រង់**String(val)****String(val,base)****String(val,decimalPlaces)****ឧទាហរណ៍**

```
String stringOne = ("Hello String");
String stringOne = String ("a");
String stringTwo = String ("this is a string");
String stringOne = String ( stringTwo + "with more");
String stringOne = String ( 13);
String stringOne = String ( analogRead( 0 ),DEC );
String stringOne = String ( 45,HEX );
String stringOne = String ( 255,BIN );
String stringOne = String ( millis( ),DEC );
String stringOne = String ( 5.698,2 );
```

▪ **charAt();**

សម្រាប់ចាប់យកតួអក្សរនៅត្រង់ Index ណាមួយ

ទម្រង់ នៃការប្រើ **string.compareTo(string2)**▪ **compareTo();**

ប្រើសម្រាប់ប្រៀបធៀប string 2

ទម្រង់**String.compareTo(string2)****Parameters**

String: a variable of type String

String2: another variable of type String

Returns

a negative number: if string comes before string2

a positive number : if string comes after string2

0: if string equals string2

- **concat();**
បង្កើត String បញ្ចូលគ្នា
ទម្រង់ **string.concat(parameter)**

ឧទាហរណ៍

```
String stringTwo = "A long integer";
StringTwo.concat( 12345 ); // "A long integer: 12345"
```

- **endsWith();**
សម្រាប់តេស្តថាតើ String បានបញ្ចប់ដោយពាក្យណាមួយឬអត់
ទម្រង់ **string.endsWith(string2)**

Parameters:

String: a variable of type String

String2: another variable of type String

Returns:

true: if string ends with the characters of string1

false: otherwise

- **equals();**
ប្រើសម្រាប់ប្រៀបធៀប String 2 ថាតើស្មើគ្នាឬអត់ ហើយអក្សរតូច និងអក្សរធំគឺខុសគ្នា។
ទម្រង់ **string.equals(string2)**

Parameters:

String,string2: variables of type String

Returns:

true: if string equals string2

false: otherwise

- **equalsIgnoreCase();**

ប្រើសម្រាប់ប្រៀបធៀប String 2 ថាតើស្មើគ្នាឬអត់ ហើយមិនគិតថាជាអក្សរធំ ឬអក្សរតូចនោះទេ។

- **indexOf();**
រក index នៃអក្សរក្នុង String ដោយរកពីមុខទៅមុន

ទម្រង់**string.indexOf(val)****string.indexOf(val, from)**

Parameters:

string: a variable of type String

val: the value to search for – char or String

from: the index to start the search from

Returns:

The index of val within the String, or – 1 if not found

▪ **lastIndexOf ();**

រក index នៃអក្សរក្នុង String ដោយរកពីក្រោយទៅមុន

ទម្រង់**string.lastIndexOf (val)****string. lastIndexOf (val, from)**

Parameters:

string: a variable of type String

val: the value to search for – char or String

from: the lastIdx to start the search from

Returns:

The lastIdx of val within the String, or – 1 if not found

▪ **length();**

សម្រាប់វាស់ចំនួនអក្សរក្នុងមួយ String

ទម្រង់**string.length()**

Parameters:

String: a variable of type String

Returns:

The length of the String in characters

- **remove();**

មានតួនាទីលុបអក្សរក្នុង String ពី Index ណាមួយរហូតដល់ទីបញ្ចប់
ទម្រង់

string.remove(index)

string.remove(index, count)

Parameters:

index: a variable of type unsigned int

count: a variable of type unsigned int

- **replace ();**

ជំនួសពាក្យណាមួយក្នុង String ដោយពាក្យថ្មី
ទម្រង់

string.replace(substring1, substring2)

Parameters:

string: a variable of type String

substring1: another variable of type String

substring2: another variable of type String

- **setCharAt();**

សម្រាប់អក្សរមួយត្រូវទៅក្នុង string នៅ index ណាមួយ
ទម្រង់

String.setCharAt(index, c)

Parameters:

string: a variable of type String

index: the index to set the character at

c: the character to store to the given location

- **startsWith();**

សម្រាប់ធ្វើតេស្តមើលថាតើពាក្យក្នុង String ចាប់ផ្តើមដោយពាក្យណាមួយឬទេ

ទម្រង់**String.startsWith(string2)**

Parameters:

string, string2: variable2 of type String

Returns:

true: if string starts with the characters of string2

false: otherwise

▪ **startsWith();**

កាត់ពាក្យពីត្រឹម index មួយទៀត

ទម្រង់**string.substring(from)****string.substring(from, to)**

Parameters:

String: a variable of type string

from: the index to start the substring at

to (optional): the index to end the substring before

Returns:

The substring

▪ **toInt ();**

បំប្លែងពីពាក្យ ឱ្យទៅជាលេខ Int

ទម្រង់**String.toInt()**

Parameters:

string: a variable of String

Returns:

1. long
2. if no valid conversion could be performed because the string doesn't start with an integer number, a zero is returned.

- **toFloat ();**

បំប្លែងពីពាក្យ ឱ្យទៅជាលេខ Float

ទម្រង់

String.toFloat()

Parameters:

String: a variable of type String

Returns:

1. Float
2. if no valid conversion could be performed because the string doesn't start with an integer number, a zero is returned.

- **toLowerCase ();**

បំប្លែងពីពាក្យដែលផ្តុំដោយអក្សរធំទៅជាអក្សរតូចទាំងអស់

ទម្រង់

string.toLowerCase()

Parameters:

string: a variable of type String

- **toUpperCase ();**

បំប្លែងពីពាក្យដែលផ្តុំដោយអក្សរតូចទៅជាអក្សរធំទាំងអស់

ទម្រង់

string.toUpperCase()

Parameters:

String: a variable of type String

- **trim ();**

កាត់ចេញនូវ Whitespace ពីមុខ និងពីក្រោយពាក្យរបស់យើង

ទម្រង់

string.trim ();

Parameters:

string: a variable of String

អនុវត្ត ចូរសរសេរកូដ ដោយបញ្ចូលទិន្នន័យជាលេខ តាមរយៈ Serial Monitor ហើយបញ្ចេញមកវិញតាម 7 segment ។

```
#define din 8
#define load 9
#define clk 10

long count;
String txt;
void setup( ) {
    pinMode( din, OUTPUT );
    pinMode( load, OUTPUT );
    pinMode( clk, OUTPUT );
    sda( 0x0c,0x01 );
    sda( 0x09,0xff );
    sda( 0x0a,0x02 );
    sda( 0x0b,0x03 );
    Serial.begin( 9600 );
}
void loop( ) {
    if( Serial.available( )>0 )
    {
        txt = Serial.readString( );
        count = txt.toInt( );
    }
    if( count < 0 )
    {
        count = 0;
    }
    if( count > 9999 )
    {
        count = 99;
    }
}
```

```

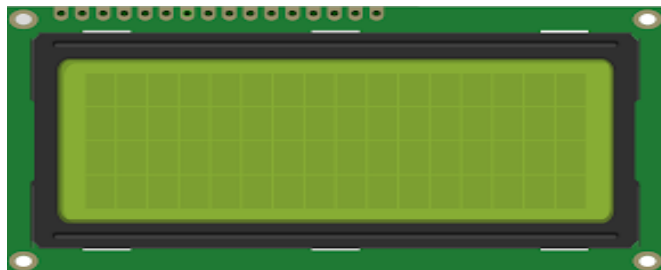
    int dig1 = count/1000;
    int dig2 = count%1000/100;
    int dig3 = count%1000%100/10;
    int dig4 = count%1000%100%10;

    sda( 1, dig1);
    sda( 2, dig2 );
    sda( 3, dig3 );
    sda( 4, dig4 );
}
void sda( byte add,byte da )
{
    digitalWrite( load, LOW );
    shiftOut( din, clk, MSBFIRST, add );
    shiftOut( din, clk, MSBFIRST, da );
    digitalWrite( load, HIGH );
}

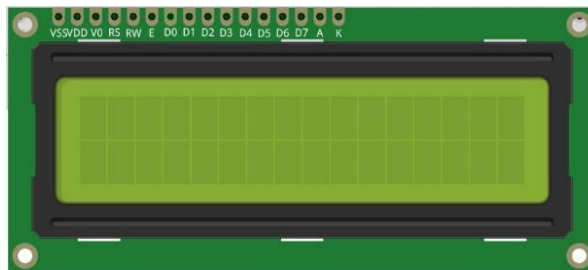
```

មេរៀនទី៦ Display LCD, OLED

ក្នុងមេរៀននេះយើងនឹងបានសិក្សាអំពីការប្រើឧបករណ៍បន្ថែមទាក់ទងនឹងការបង្ហាញចេញនូវទិន្នន័យមួយចំនួនឱ្យយើងមើល ជំនួស 7 segment និងចាប់ផ្តើមប្រើ Library ផ្សេងៗដទៃទៀត។



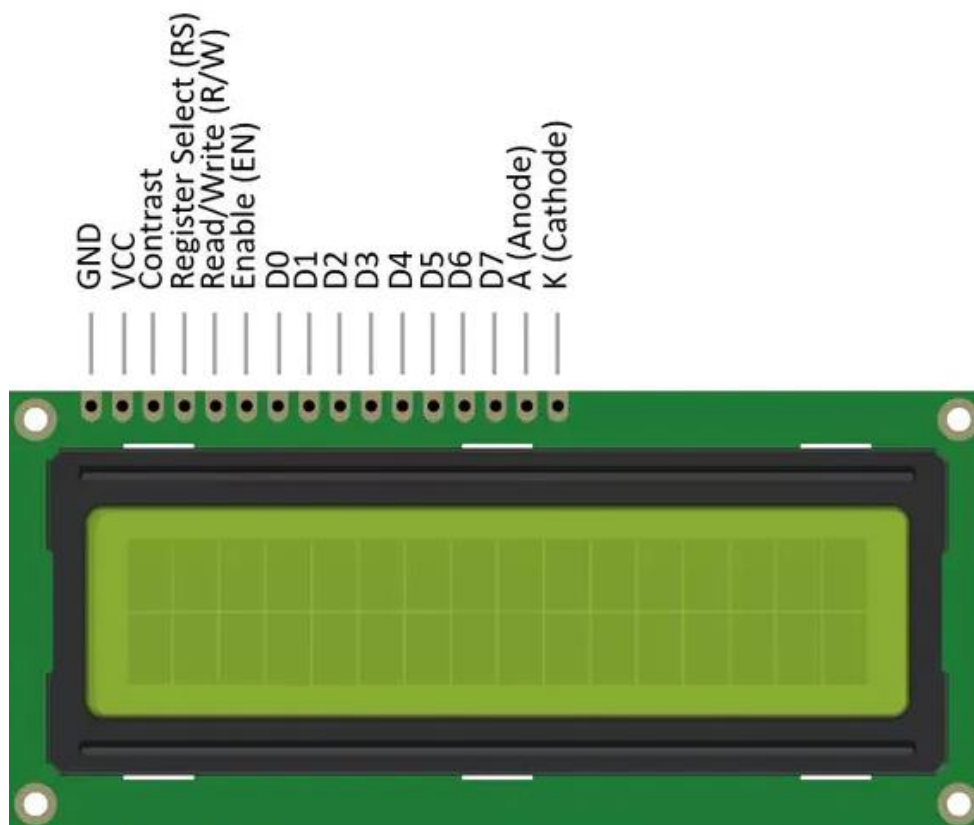
រូបទី៦.១ LCD 20x4



រូបទី៦.២ LCD 16x2

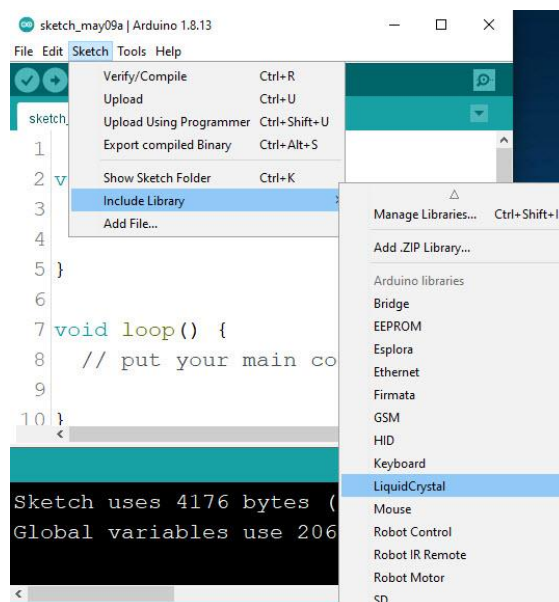
១. Liquid Crystal Display (LCD)

LCD ជាឧបករណ៍អេឡិចត្រូនិចមួយប្រភេទដែលត្រូវបានគេប្រើសម្រាប់បង្ហាញព័ត៌មានណាមួយរបស់កូដតាមដែលយើងកំណត់ ហើយ LCD មានភាពច្បាស់លាស់ និងលម្អិតជាងអង្កត់លេខព្រោះអាចបង្ហាញជាតួអក្សរឡាតាំងស្ទើរគ្រប់អក្សរ និងនិមិត្តសញ្ញាផ្សេងៗ។ LCD មានច្រើនទំហំ ហើយទំហំដែលប្រើទូទៅច្រើនជាងគេគឺ 16x2 និង 20x4 ដោយក្នុងមេរៀននេះ យើងសិក្សាជាមួយ LCD 16x2 ។ លេខ 16 ជាចំនួនអក្សរដែលអាចសរសេរបាន ចំណែកលេខ 2 ជាចំនួនជួរអក្សរ។ ដើម្បីរបស់ LCD 16x2 ទាំងអស់មាន ១៦ជើងដូចក្នុងរូបទី៦.៣ ។



រូបទី៦.៣ LCD 16x2 មានឈ្មោះជើង

ដើម្បីអាចសរសេរកូដបញ្ជា LCD បានយើងត្រូវបញ្ចូល Library ជាមុនសិន ដោយចូលទៅកាន់ Menu Sketch >> Include Library >> LiquidCrystal ពេលនោះយើងនឹងឃើញនៅក្នុង Sketch មាន #include <LiquidCrystal.h> តាមរូបទី៦.៤ ។



រូបទី៦.៤ ការទាញយក Library មកប្រើ

ឧទាហរណ៍ ចូរសរសេរកូដបញ្ជាឱ្យ LCD ចេញពាក្យ "Hello KCIT" ។

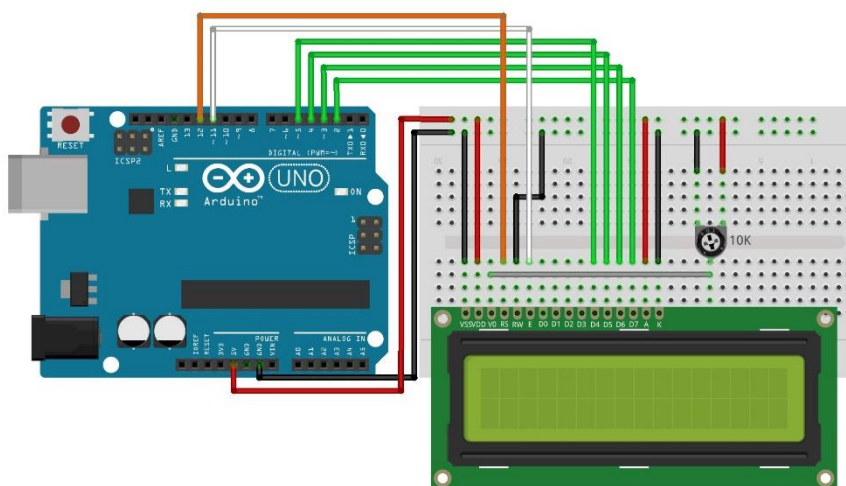
```
#include <LiquidCrystal.h>

LiquidCrystal lcd( 12,11,5,4,3,2);

void setup( ) {
    lcd.begin( 16,2);
}

void loop( ) {
    lcd.print( "Hello KCIT" );
    lcd.setCursor( 0,1);
}
```

ខាងក្រោមជារូបរៀបត LCD និង board Arduino



រូបទី៦.៥ ការតភ្ជាប់ LCD ចូលជាមួយ board Arduino

Function របស់ Liquid Crystal មានដូចតទៅនេះ

- **LiquidCrystal();**

ប្រើសម្រាប់ប្រកាសបើកដើម្បី Arduino ឱ្យ LCD

ទម្រង់

LiquidCrystal(rs, enable, d4, d5, d6, d7)

LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)

LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)

LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)

ឧទាហរណ៍

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd( 1,2,3,4,5,6 );
```

```
void setup( ) {
```

```
    lcd.begin( 16,2 );
```

```
    lcd.print( "BT.Sound" );
```

```
}
```

```
void loop( ) {
```

```
}
```

- **begin()**

ប្រើសម្រាប់ឱ្យ Library ចាប់ផ្តើមដំណើរការ

ទម្រង់

Lcd.begin(cols, rows);

Parameters:

lcd: a variable of type LiquidCrystal

cols: the number of columns that the display has

rows: the number of rows that the display has

- **clear()** ប្រើសម្រាប់លុបអក្សរទាំងអស់នៅលើ Screen។

- **home()** ទាញ Cursor ទៅកាន់ Colum = 0, Row = 0។

- **setCursor()** ដាក់ Cursor ទៅកាន់ Colum, Row ណាមួយ។

ទម្រង់ lcd.setCursor(col, row)

- write()

សម្រាប់បញ្ជូនអក្សរមួយតួទៅកាន់ LCD។

ទម្រង់ lcd.write(data)

ឧទាហរណ៍

```
#include <LiquidCrystal.h>
LiquidCrystal lcd( 1,2,3,4,5,6 );
void setup( ) {
    lcd.begin( 16,2 );
    lcd.write( "A" );
}
void loop( ) {
}
```

- print()

មានតួតួនាទីបញ្ជូនពាក្យដែលយើងសរសេរចេញទៅ LCD ។

ទម្រង់

lcd.print(data)

lcd.print(data, BASE)

ឧទាហរណ៍

```
lcd.print( "Hello" );
lcd.print( 73,HEX );
```

- cursor() សម្រាប់បង្ហាញ Cursor នៅលើ Screen។
- noCursor() មិនបង្ហាញ Cursor នៅលើ Screen។
- blink() សម្រាប់ឱ្យ Cursor លោតភ្លឺបន្តិច។
- noBlink() បិទមិនឱ្យ Cursor លោតភ្លឺបន្តិច។
- display មានតួតួនាទីសម្រាប់ឱ្យ LCD បង្ហាញអក្សរ។
- noDisplay() មានតួតួនាទីសម្រាប់លាក់មិនឱ្យ LCD បង្ហាញអក្សរ តែមិនមែនលុបទាំងស្រុងទេ គឺអាចបង្ហាញចេញមកវិញតាមរយៈ display()។
- scrollDisplayLeft() មានតួតួនាទីសម្រាប់រុញអក្សរដែលមានលើ Screen ទាំងអស់ទៅខាងធ្វេងមួយតួអក្សរ។

- `scrollDisplayRight( )` មានតួនាទីសម្រាប់រុញអក្សរដែលមានលើ Screen ទាំងអស់ទៅខាងស្តាំមួយតួអក្សរ។
- `autoscroll( )` នៅពេលដែលអក្សរសរសេរលើសចំនួន Column នោះវានឹងរំកិលអក្សរមួយទៅឆ្វេងឬទៅស្តាំ។
- `noAutoscroll( )` នៅពេលដែលអក្សរសរសេរលើសចំនួន Column នោះវានឹងមិនរំកិលអក្សរមួយទៅ ឆ្វេងឬទៅស្តាំនោះទេ ។
- `leftToRight( )` មានតួនាទីកំណត់យកទិសដៅអក្សរចេញពីឆ្វេងទៅ ស្តាំ។
- `rightToLeft( )` មានតួនាទីកំណត់យកទិសដៅអក្សរចេញពីស្តាំទៅ ឆ្វេង។
- `createChar( )` សម្រាប់បង្កើតនិមិត្តសញ្ញា ឬតួអក្សរថ្មីណាមួយដែលមិនមែនជាអក្សរឡាតាំង

ទម្រង់ `Lcd.createChar( num, data )`

Parameters:

`lcd`: a variable of type `LiquidCrystal`

`num`: which character to create (0 to 7)

`data`: the character's pixel data

ឧទាហរណ៍១

```
#include <LiquidCrystal.h>
LiquidCrystal lcd( 1,2,3,4,5,6 );
byte smiley[8] ={
    B00000,
    B10001,
    B00000,
    B00000,
    B10001,
    B01110,
    B00000,
```

```

    };

void setup( ) {
    lcd.createChar( 0, smiley );
    lcd.begin( 16,2 );
    lcd.write( byte( 0 ) );
}

void loop( ) {
}

```

ឧទាហរណ៍២ ចូរសរសេរកូដមួយ ដោយយើងបញ្ចូលពាក្យមួយចំនួនតាម Serial Monitor ហើយបង្ហាញនៅលើ LCD។

```

#include <LiquidCrystal.h>
LiquidCrystal lcd( 1,2,3,4,5,6 );
void setup( ) {

    lcd.begin( 16,2 );
    lcd.write( byte( 0 ) );
}

void loop( ) {
    if ( Serial.available( ) > 0 ){
        String txt = Serial.readString( );
        lcd.clear( );
        lcd.setCursor( 0,0 );
        lcd.print( txt );
    }
}

```

ឧទាហរណ៍៣ ចូរសរសេរកូដបង្កើតជានាឡិការឌីជីថល បង្ហាញនៅលើ LCD

```

#include <LiquidCrystal.h>
LiquidCrystal lcd( 1,2,3,4,5,6 );
char tim[9];

```

```

void setup( ) {
    lcd.begin( 16,2 );
}
void loop( ) {
    for( long i=0; i<86400;i++){
        int h = i/ 3600;
        int m = i%3600/60;
        int s = i%3600%60;
        sprintf( tim,"%02d:%02d:%02d",h,m,s);
        lcd.clear( );
        lcd.setCursor( 0,0);
        lcd.print( String( tim ) );
        delay( 1000 );
    }
}

```

ឧទាហរណ៍៤ ចូរសរសេរកូដមួយដែលមានកុងតាក់ចុច Up / Down និង Shift Left/ Right ព្រមទាំងមានកុងតាក់ Start/ Reset ផង។ ឧទាហរណ៍ នេះគឺយើងធ្វើប្រព័ន្ធមួយដែលមាន Timer ហើយធ្វើការបញ្ចូលតម្លៃតាមរយៈការ Set Up/ Down រួចបង្ហាញលេខលើ LCD ។ ដើម្បីអាចបញ្ចូលបានលឿនយើងអាចប្រើ Shift Left/ Right។ នៅពេលចុចកុងតាក់ Start គឺចាប់ផ្តើមរាប់លេខ ចុះរហូតដល់សូន្យ ហើយពេលចុះដល់លេខសូន្យនឹងចេញពាក្យ Alarm នៅក្រោម Process time ត្រង់ Timer count down រហូតដល់យើងចុចកុងតាក់ Set ជាប់ទើបបាត់ Alarm ។ គួរបញ្ជាក់ថា Up = (A1), Down = (A2), Left = (A3), Right = (A4), Start = (A5), Set = (A0) ។

```

#include <LiquidCrystal.h>
#include <LiquidCrystal.h>
#define up digitalRead( 1 )
#define down digitalRead( 6 )
#define L digitalRead( 7 )
#define R digitalRead( 8 )
#define sta digitalRead( 10 )
#define se digitalRead( 13 )
LiquidCrystal lcd( 12, 11, 5, 4, 3, 2 );

```

```

char time[9];
int pos, num;
void setup( ) {
    lcd.begin( 16, 2 );
    pinMode( up, INPUT );
    pinMode( down, INPUT );
    pinMode( L, INPUT );
    pinMode( R, INPUT );
    pinMode( sta, INPUT );
    pinMode( se, INPUT );
    distxt( 7, 0, "||" );
    distxt( 7, 1, "||" );
    distxt( 2, 0, "SET" );
    distxt( 11, 0, "PRO" );
    distxt( 1, 1, dis ( num ) );
    distxt( 10, 1, dis( num ) );
}

void loop( ) {
    if ( set==1 ){
        delay( 200 );
        distxt( 10, 1, dis( 0 ) );
        pos = 5;
        lcd.cursor( );
        lcd.blink( );
        lcd.setCursor( pos,1 );
    }

    if( sta==1 ){
        delay( 200 );
        lcd.noCursor( );
        lcd.noBlink( );
        for( int i = num; i > -1; i-- ){

```

```

        distxt( 10, 1, dis(i) );
        delay( 1000 );
        if( set==1 ){
            break;
        }
    }

while( set==0 ){
    distxt( 10, 1, "Alarm" );
    delay( 1000 );
    distxt( 10, 1, "    " );
    delay( 1000 );
}

if ( |== 1 ){
    delay( 200 );
    pos--;
    if( pos==3 ){
        pos=2;
    }
    if( pos==0 ){
        pos= 5;
    }
    lcd.setCursor( pos,1 );
}

if ( |== 1 ){
    delay( 200 );
    pos++;
    if( pos==3 ){
        pos=4;
    }
    if( pos==6 ){

```

```

        pos= 1;
    }
    lcd.setCursor( pos,1 );
}

if( up==1 ){
    delay( 200 );
    switch( pos ){
        case 5:
            num++;
            break;
        case 4:
            num += 10;
            break;
        case 2:
            num += 60;
            break;
        case 1:
            num += 600;
            break;
    }
    if ( num> 5999 ){
        num = 0;
    }
    distxt( 1,1,dis( num ) );
    lcd.setCursor( pos,1 );
}

if ( down==1 ){
    delay( 200 );
    switch( pos ){
        case 5:
            num--;

```

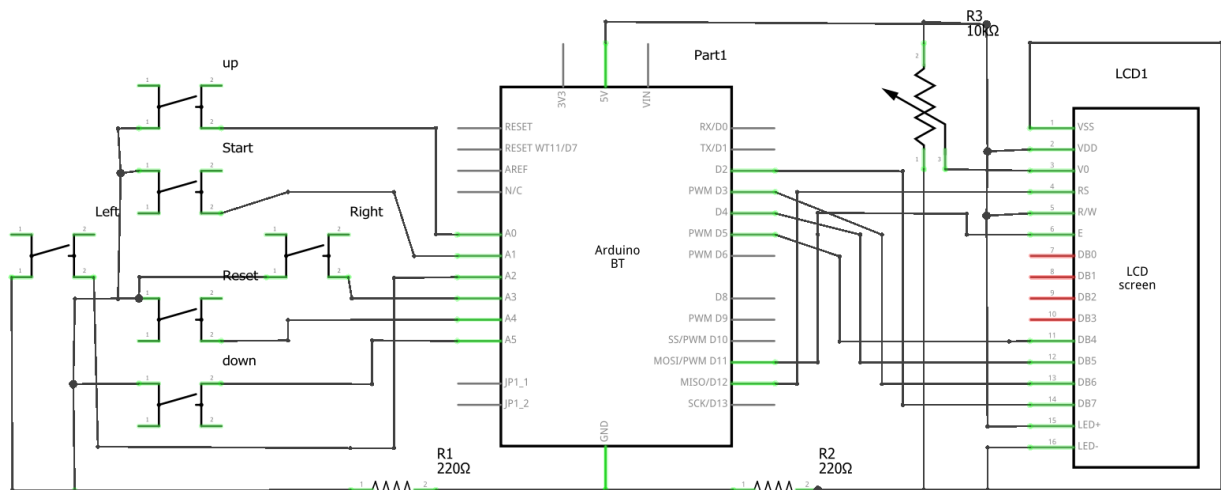
```

        break;
    case 4:
        num-=10;
        break;
    case 2:
        num-=60;
        break;
    case 1:
        num-=600;
        break;
    }
    if( num < 0){
        num = 0;
    }
    distxt( 1,1,dis( num ) );
    lcd.setCursor( pos,1 );
}

void distxt( int col, int row, String txt ) {
    lcd.setCursor( col, row );
    lcd.print( txt );
}

String dis( int c ){
    int m = c/60;
    int s = c%60;
    sprintf( tim, "%02d:%02d", m, s );
    return String( tim );
}
}

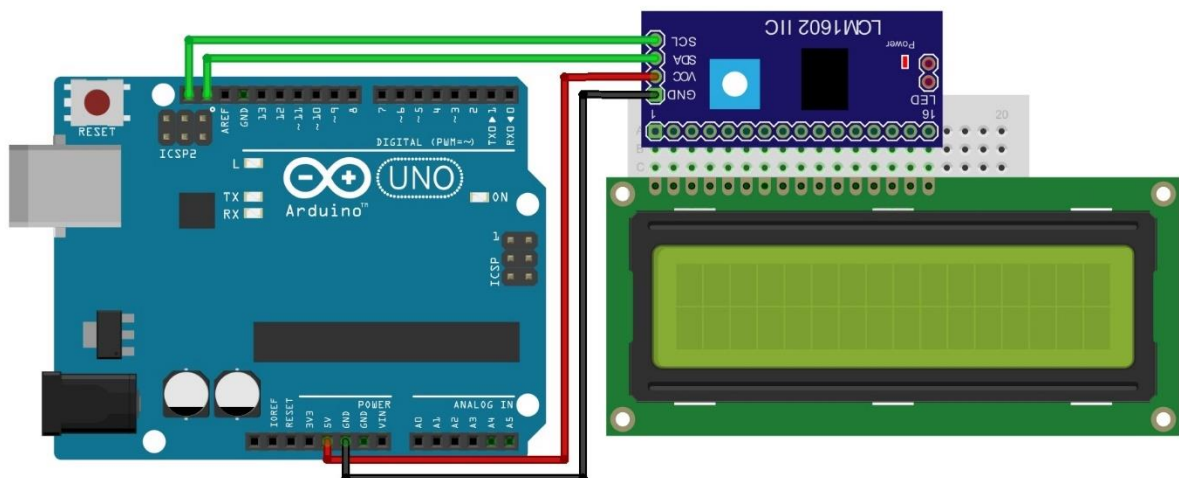
```



រូបទី៦.៦ ការតភ្ជាប់ត្រីកូដឧទាហរណ៍

ក. ការប្រើ LCD I2C

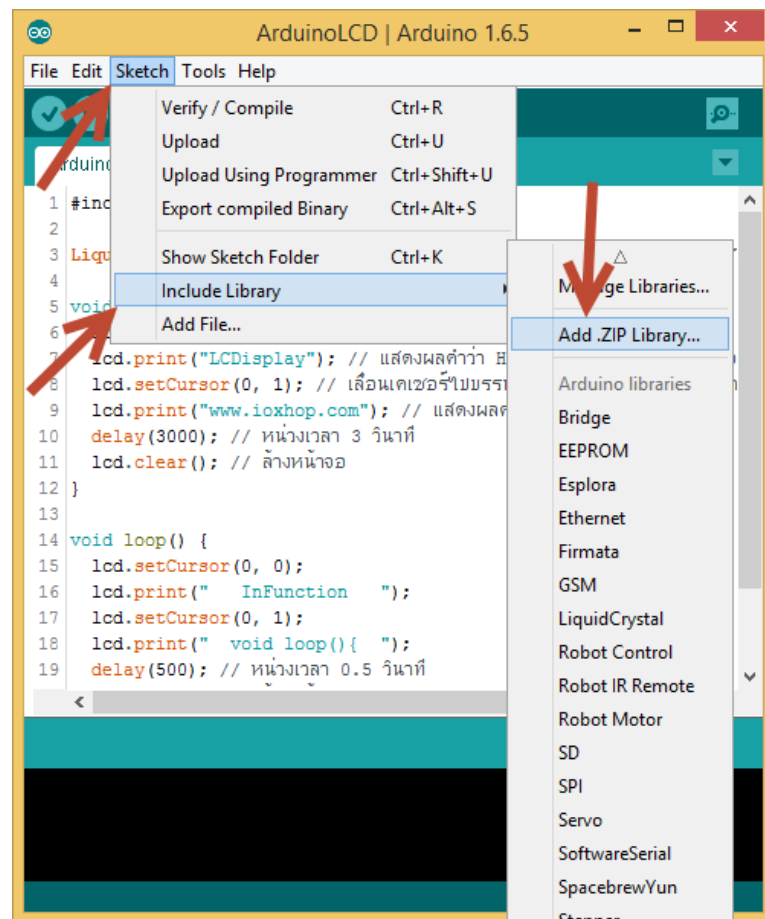
I2C គឺការតភ្ជាប់ LCD ជាខ្នង ហើយត្រូវប្រើ Board I2C មកភ្ជាប់ជា LCD នឹង ប្រើខ្សែតែ៤ ចេញមកតាមមួយ board Arduino ។ ខ្សែដែលតភ្ជាប់មកពី I2C មានខ្សែត្រូវតែចូលជើងភ្លើងនឹងម៉ាស់ ហើយពីរទៀតតែចូលជើង SDA, SCL របស់ board Arduino ដូចក្នុងរូបទី ៦.៧ ។ គេបង្កើត Board I2C នេះឡើងគោលបំណងសំខាន់ គឺកាត់បន្ថយការប្រើខ្សែច្រើន សំបុក។



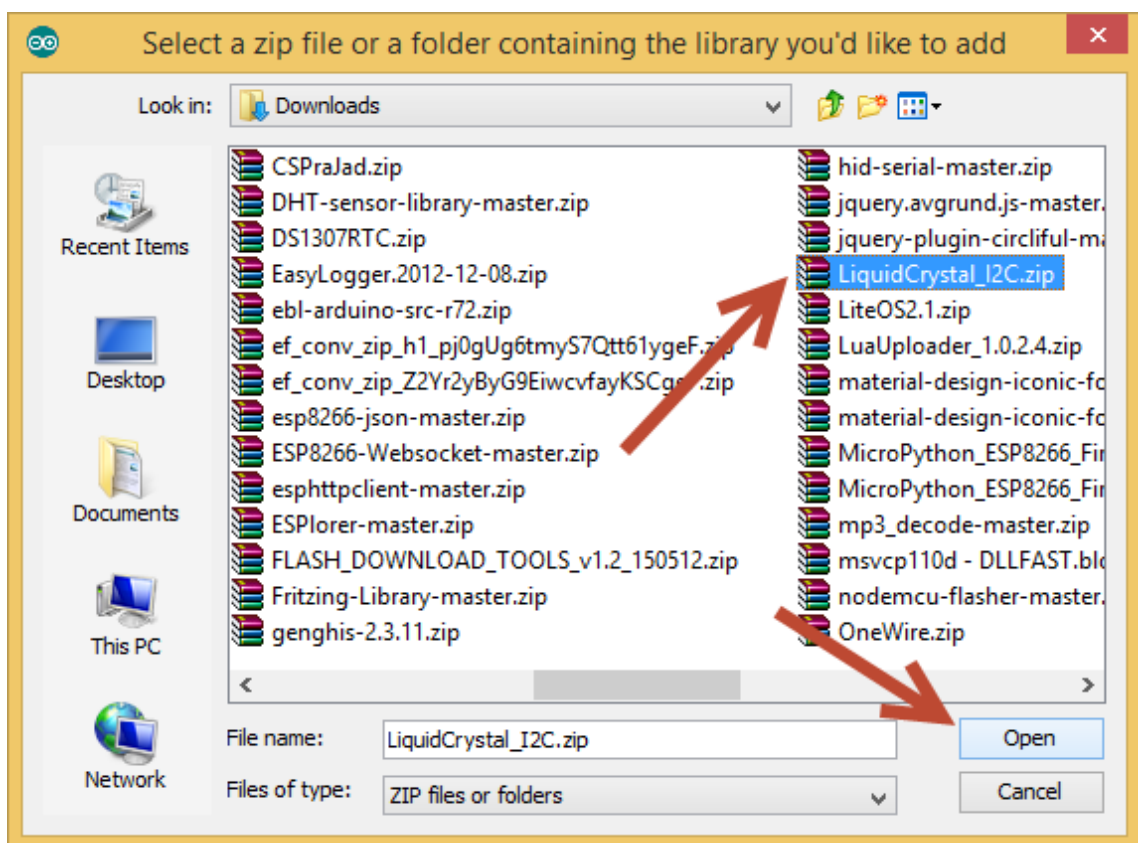
រូបទី៦.៧ ការតភ្ជាប់ Board I2C ជាមួយ Arduino

ដើម្បីប្រើ Board I2C នេះជាមួយ Arduino បាន ត្រូវទាញយក Library ថ្មីមួយទៀតឈ្មោះថា LiquidCrystal_I2C ដោយទាញយកតាមវិធីក្នុងរូបទី ៦.៨ និងរូបទី ៦.៩ ខាងក្រោម ៖

ឧទាហរណ៍ ការប្រើ Board I2C ជាមួយ Arduino



រូបទី៦.៨ របៀបចូលទាញយក Library



រូបទី៦.៩ របៀបចូលទាញយក Library

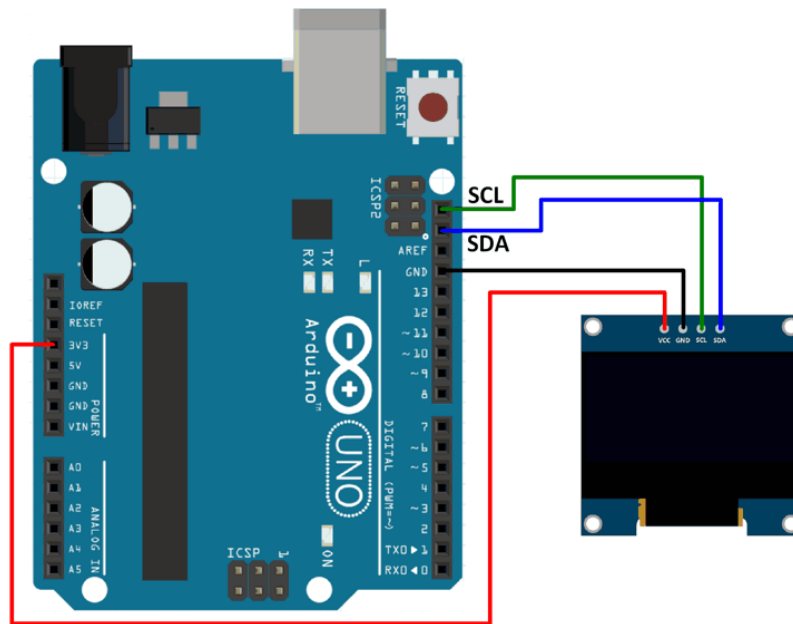
```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd( 0x3F, 16, 2 );
void setup( )
{
  lcd.begin( 16, 2 );
  lcd.print( "Hello KCIT" );
  lcd.setCursor( 0, 1 );
  lcd.print( "BT.Sound" );
  delay( 3000 );
  lcd.clear( );
}

void loop( )
{
  lcd.setCursor( 3, 0 );
  lcd.print( " Wellcome " );
  lcd.setCursor( 2, 1 );
  lcd.print( "KompongThom" );
  delay( 500 );
  lcd.clear( );
  delay( 500 );
}

```

២. Graphic OLED ជាប្រភេទ LCD ខ្នាតតូច ដែលងាយស្រួលប្រើសម្រាប់ទីតាំងតូចចង្អៀត ។ ម្យ៉ាងទៀតអាចបញ្ចេញរូបភាពបានច្រើន មិន LiquidCrystal ទេ។ តាមរយៈ OLED យើងអាចបង្ហាញព័ត៌មានបានកាន់តែច្រើន ។

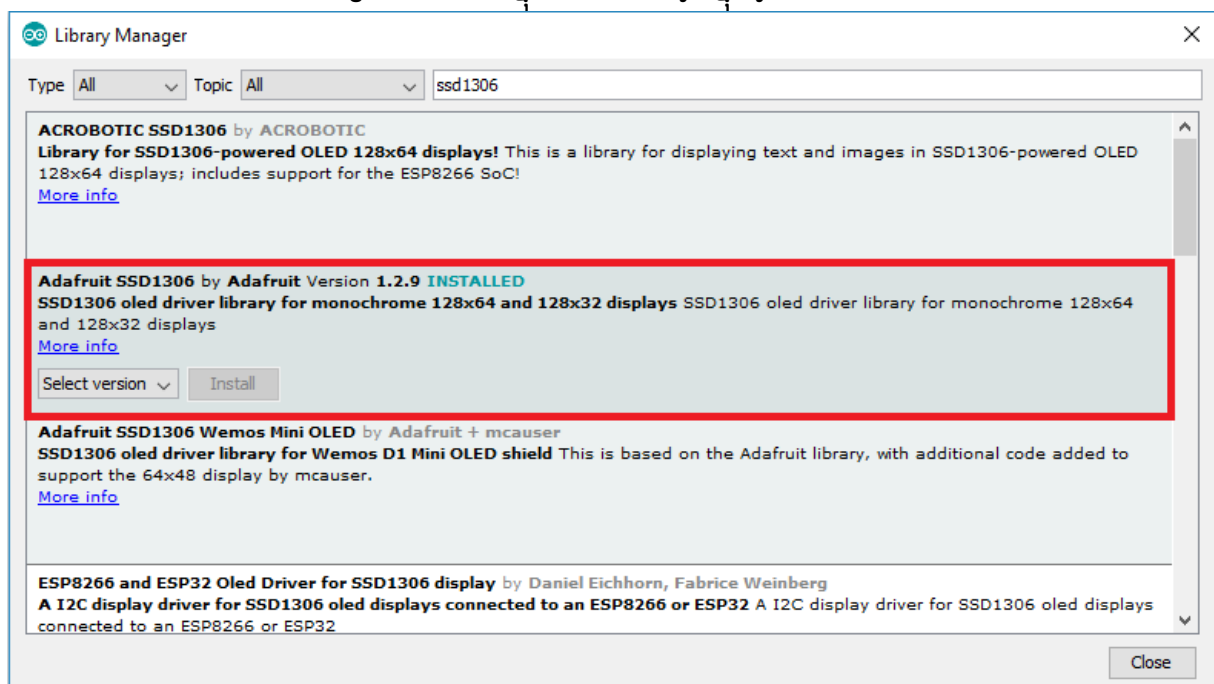


រូបទី៦.១០ ការភ្ជាប់ OLED ជាមួយ board Arduino

OLED មិនខុសពី LCD នោះទេ ដើម្បីអាចប្រើវាបានយើងត្រូវមាន Library មួយចំនួនដូចខាងក្រោម Adafruit_GFX.h, gfxfont.h, Adafruit SSD1306.h។ ដើម្បីទាញយក Library ទាំងអស់នោះមកប្រើបានយើងអាចធ្វើតាមជំហានដូចខាងក្រោម៖

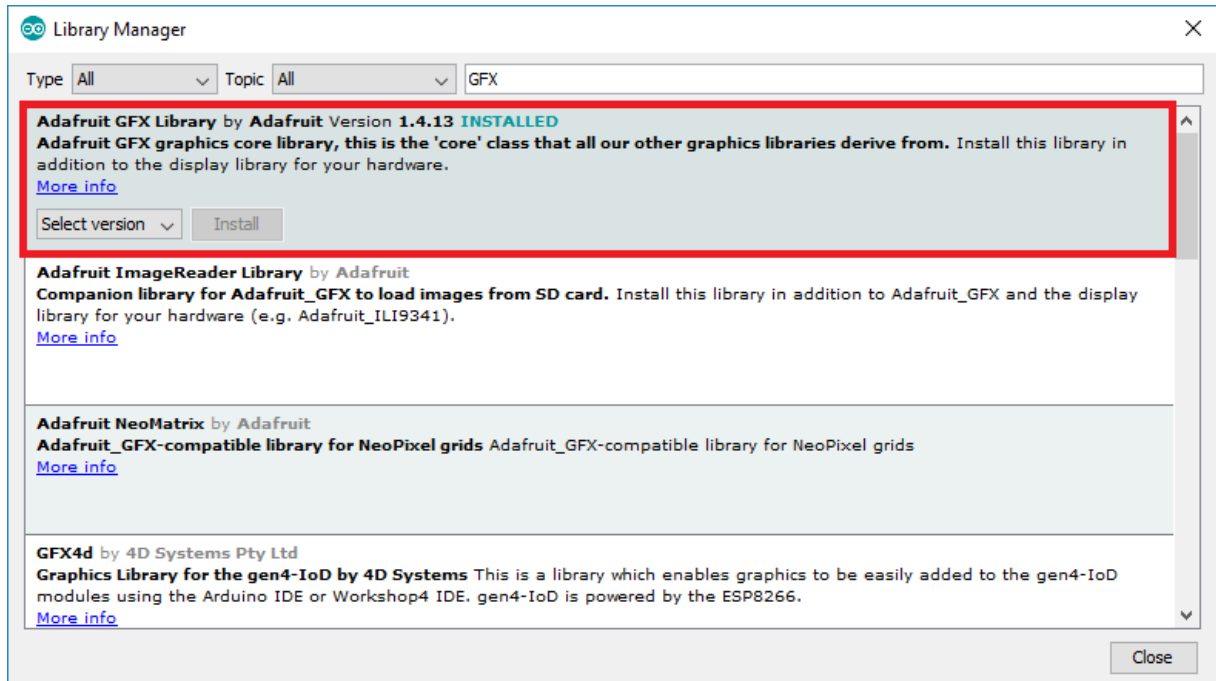
ជំហានទី១ ចុចលើ Sketch > include Library > Manage Libraries

ជំហានទី២ សរសេរពាក្យ SSD1306 ក្នុង search ដូចក្នុងរូបទី៦.១១



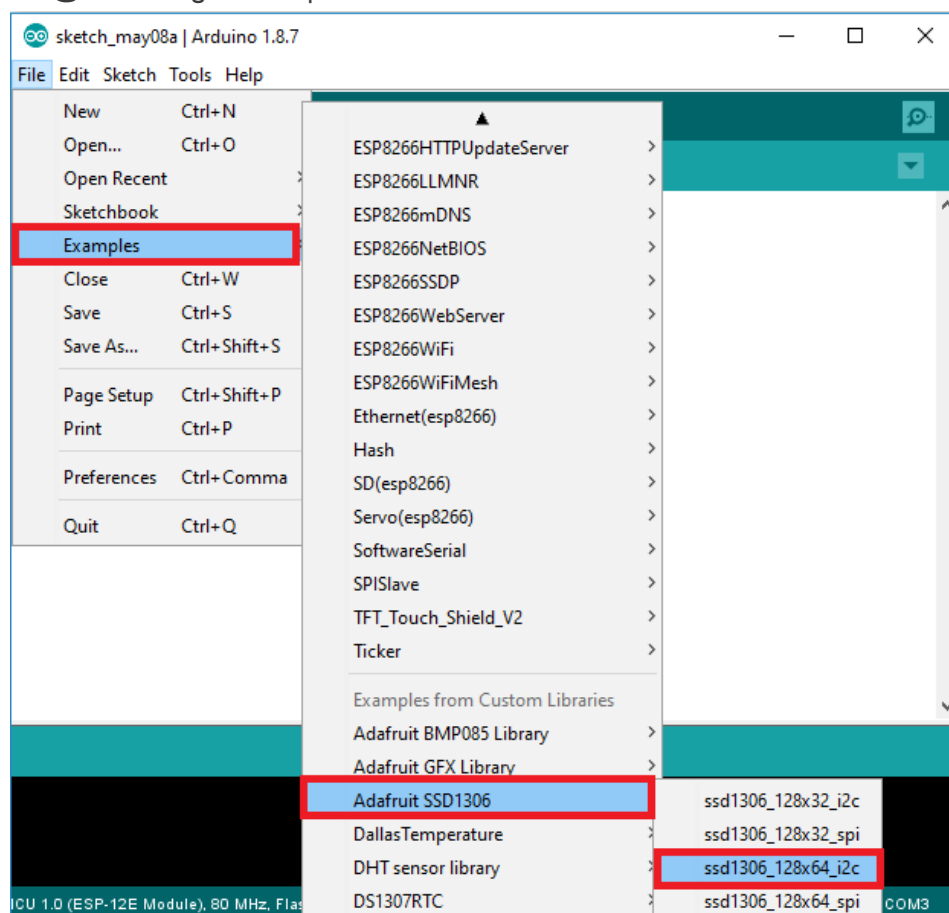
រូបទី៦.១១ ការទាញយក SSD1306

ជំហានទី៣ បន្ទាប់ពីទាញយក SSD1306 បានហើយ ត្រូវទាញយក GFX មួយទៀតដូចរូបទី ៦.១២



រូបទី៦.១២ ការទាញយក GFX

ជំហានទី៤ ក្រោយពីដំឡើង Library ទាំងពីររួចហើយត្រូវ restart កម្មវិធី Arduino IDE ម្តងៗ បន្ទាប់មកចូលទាញយកមកប្រើដោយចុច File > Examples > Adafruit SSD1306 ។



រូបទី៦.១៣ ការចូលទាញយក Library មកប្រើ

ក. ទម្រង់ដែលត្រូវប្រើជាមួយ OLED

- Adafruit_SSD1306(); ប្រើសម្រាប់ប្រកាសអញ្ញាតរបស់ Library នេះ

ទម្រង់

Adafruit_SSD1306 OLED(RS_pin)

Parameters

RS_pin: reset pin

RS_pin = LED_BUILTIN: យើងប្រើក្នុងករណីគ្មានជើង Reset ពីខាងក្រៅ

- begin(); ប្រើសម្រាប់ផ្ដើមដំណើរការឱ្យ OLED នៅ address របស់វា

ទម្រង់

OLED .begin(address)

Parameters

Address: address របស់ OLED ក្នុង Bus i2C

Return

Logo Adafruit

- Display(); សម្រាប់បញ្ជាឱ្យ OLED បង្ហាញអក្សរ ឬ រូបរាងចេញមក

ទម្រង់

OLED.display()

- clearDisplay(); សម្រាប់បញ្ជាឱ្យ OLED សម្អាតអក្សរ ឬរូបចេញ

ទម្រង់

OLED.drawPixel(x, y, color)

Parameters

x: x position horizontal

y: y position vertical

color: WHITE/BLACK

- drawCircle(); សម្រាប់គូសរង្វង់មួយមានផ្ចិតនៅត្រង់ទីតាំង x, y មានកាំ r

ទម្រង់

OLED.drawCircle(x, y, r, color)

Parameters

x: x position horizontal

y: y position vertical

r: radius

color: WHITE/BLACK

- **drawRect()**: សម្រាប់គូសចតុកោណកែងពេញនៅត្រង់ទីតាំង x,y មានបណ្តោយ w និងទទឹង h
ទម្រង់

OLED.drawRect(x, y, w, h, color)

Parameters

x: x position horizontal

y: y position vertical

w: width

h: height

color: WHITE/BLACK

- **fillRect()**: សម្រាប់គូសចតុកោណកែងមានលុបជ្រុងកោងនៅត្រង់ទីតាំង x, y មាន
បណ្តោយ w និងទទឹង h
ទម្រង់

OLED.fillRect(x, y, w, h, color)

Parameters

x: x position horizontal

y: y position vertical

w: width

h: height

color: WHITE/BLACK

- **drawRoundRect()**: សម្រាប់គូសចតុកោណកែង មានលុបជ្រុងកោងនៅត្រង់ទីតាំង x, y មាន
បណ្តោយ w និងទទឹង h ។
ទម្រង់

OLED.drawRoundRect(x, y, w, h, r, color)

Parameters

x: x position horizontal

y: y position vertical

w: width

h: height

r: radius filler

color: WHITE/BLACK

- `fillRoundRect( );` សម្រាប់គូសចតុកោណកែងមានលុបជ្រុងកោងនៅត្រង់ទីតាំង x, y មានបណ្តោយ w និងទទឹង h ។

ទម្រង់

`OLED.fillRoundRect(x, y, w, h, r, color)`

Parameters

x: x position horizontal

y: y position vertical

w: width

h: height

r: radius filler

color: WHITE/BLACK

- `drawTriangle( );` សម្រាប់គូសត្រីកោណចេញពីបីចំណុច

ទម្រង់

`OLED.drawTriangle(x1, y1, x2, y2, x3, y3, color)`

- `fillTriangle( );` សម្រាប់គូសត្រីកោណចេញពីបីចំណុច

ទម្រង់

`OLED.drawTriangle(x1, y1, x2, y2, x3, y3, color)`

- `drawLine( );` សម្រាប់គូសបន្ទាត់

ទម្រង់

`OLED.drawLine(x1, y1, x2, y2, color)`

- `setTextSize( );` សម្រាប់កំណត់ទំហំអក្សរ

ទម្រង់

`OLED.setTextSize(size)`

- `setTextColor( );` សម្រាប់កំណត់ពណ៌អក្សរ

ទម្រង់

`OLED.setTextColor( color )`

Parameters

color: WHITE/BLACK

- `setCursor( );` សម្រាប់កំណត់ទីតាំង Cursor

ទម្រង់

`OLED.setCursor(x, y)`

- `print/println( );` សម្រាប់សរសេរអក្សរ

ទម្រង់

`OLED.print( string )`

`OLED.println( string )`

`OLED.print( val,BASE )`

ខ. ដាក់រូបភាព OLED

- `drawBitmap( );` សម្រាប់ដាក់រូបទៅក្នុង OLED

ទម្រង់

`OLED.drawBitmap(x, y, bitmap, w, h, color)`

ឧទាហរណ៍

```
#include <Adafruit_GFX.h>
```

```
#include <gfont.h>
```

```
#include <Adafruit_SSD1306.h>
```

```
Adafruit_SSD1306 oled( 13 );
```

```
// '1111', 14x22px
```

```
const unsigned char myBitmap [] PROGMEM = {
```

```
    0xff, 0xfc, 0xff, 0xfc,
```

```
    0xff, 0xfc, 0xff, 0x1c,
```

```
    0xfe, 0x1c, 0xfe, 0x1c,
```

```

    0xf8, 0x1c, 0xf0, 0x1c,
    0xf3, 0x1c, 0xff, 0x1c,
    0xff, 0x1c, 0xff, 0x1c,
    0xff, 0x1c, 0xff, 0x1c,
    0xff, 0x1c, 0xff, 0x1c,
    0xff, 0x1c, 0xff, 0x1c,
    0xff, 0x1c, 0xff, 0x1c,
    0xff, 0xfc, 0xff, 0xfc
};

void setup( )
{
    oled.begin( 0x3c );
    oled.clearDisplay( );
    oled.drawBitmap( 0, 0, myBitmap, 25, 25, WHITE );
    oled.display( );
    delay( 1000 );
}

void loop( ) {

```

ដើម្បីអាចរករូបភាពទៅជាកូដបាន យើងអាចចូលទៅគេហទំព័រ

<https://javl.github.io/image2cpp/> ពេលចូលដល់ក្នុងគេហទំព័រ ចុចលើពាក្យ Choose File បន្ទាប់មករើសរូបភាពពីក្នុងកុំព្យូទ័រ រូបដែលយើងចង់ប្តូរ (រូបមានទំហំមិនលើស 128x 64 និងជារូបភាពសង្ខេប) ។ នៅត្រង់ Code output រើសយក Arduino Code ហើយប្តូរឈ្មោះតាមរយៈ identifier ហើយចុងក្រោយចុចលើ Generate code ដើម្បីបាន Array of image ចុចរួចនឹងបានកូដ Arduino ហើយយើងគ្រាន់តែ copy code យកមកប្រើតាមដែលយើងត្រូវការ។

1. Select image

Choose Files No file chosen

or

1. Paste byte array



128 x 64 px

Read as horizontal

Read as vertical

2. Image Settings

Canvas size(s):

No files selected

Background color:

☒ White ☐ Black ☐ Transparent

Invert image colors

☐

Brightness / alpha threshold:

128

0 - 255; If the brightness of a pixel is above the given level the pixel becomes white, otherwise they become black. When using alpha, opaque and transparent are used instead.

Scaling

original size

Center:

☐ horizontally ☐ vertically

Note: centering the image only works when using a canvas larger than the original image.

3. Preview

4. Output

Code output format

plain bytes

Draw mode:

Horizontal - 1 bit per pixel

If your image looks all messed up on your display, like the image below, try using a different mode.



Generate code



ឧទាហរណ៍: ចូរសរសេរកូដដែលយើងបញ្ចូលពាក្យពី Serial Monitor ហើយបង្ហាញនៅលើ OLED

```
#include <Adafruit_GFX.h>
```

```
#include <gfont.h>
```

```

#include <Adafruit_SSD1306.h>

Adafruit_SSD1306 oled( LED_BUILTIN );

void setup( ){
    Serial.begin( 9600 );
    oled.begin( 0x3c );
    oled.clearDisplay( );
    oled.display( );
}

void loop( ){
    if ( Serial.available( ) > 0 )
    {
        String txt = Serial.readString( );
        oled.clearDisplay( );
        oled.setCursor( 0,0 );
        oled.setTextSize( 1 );
        oled.print( txt );
        oled.display( );
    }
}

```

ឧទាហរណ៍ ចូរសរសេរកូដមួយដែលមានរូបចេញលើ OLED (រូបអ្វីក៏បានដែរ) ។

```

#include <Adafruit_GFX.h>
#include <gfxfont.h>
#include <Adafruit_SSD1306.h>
Adafruit_SSD1306 oled( 13 );
// 'BT.Sound' 120x110px
const unsigned char myBTSound [] PROGMEM = {
    0x9f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x1f,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x3f, 0xff,
    0xff, 0xff, 0xff, 0xff, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xe0, 0x00, 0x00, 0x00, 0x03, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00,

```

0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x01, 0xbf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x4f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x02, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f,
 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x2f, 0xff,
 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xff, 0xff,
 0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xff, 0xff, 0xff,
 0xff, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff,
 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x81, 0xff, 0xff, 0xff, 0xe0,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xc1, 0xff, 0xff, 0xff, 0xc0, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xc0, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xe0, 0x7f, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0xc0, 0x7f, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x7f, 0xc0, 0x3f, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x01, 0xff, 0x80, 0x1f, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x03, 0xff, 0x00, 0x1f, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x07, 0xfe, 0x00, 0x0f, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x0f, 0xfc, 0x00, 0x00, 0x00, 0x0f,
 0xfc, 0x00, 0x0f, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x3f, 0xff, 0x00, 0x00, 0x00, 0x3f, 0xf8,
 0x00, 0x07, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0xff, 0xff, 0xc0, 0x00, 0x00, 0x7f, 0xf0, 0x00,
 0x03, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x01, 0xff, 0xff, 0xe0, 0x00, 0x01, 0xff, 0xe0, 0x00, 0x03,
 0xff, 0xff, 0xc0, 0x00, 0x00, 0x07, 0xff, 0xff, 0xf0, 0x00, 0x03, 0xff, 0x80, 0x00, 0x01, 0xff,
 0xff, 0xc0, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xf0, 0x00, 0x07, 0xff, 0x00, 0x00, 0x01, 0xff, 0xff,
 0x80, 0x00, 0x00, 0x1f, 0xff, 0xff, 0xf8, 0x00, 0x1f, 0xfe, 0x00, 0x00, 0x00, 0xff, 0xff, 0x80,
 0x00, 0x00, 0x3f, 0xf8, 0x07, 0xf8, 0x00, 0x3f, 0xf8, 0x00, 0x00, 0x00, 0xff, 0xff, 0x00, 0x00,
 0x00, 0x7f, 0xe0, 0x03, 0xfc, 0x00, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x7f, 0xff, 0x00, 0x00, 0x00,
 0xff, 0xc0, 0x01, 0xfc, 0x03, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x7f, 0xff, 0x00, 0x00, 0x00, 0xff,
 0x80, 0x00, 0xfc, 0x07, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x7f, 0xfe, 0x00, 0x00, 0x01, 0xff, 0x00,
 0x00, 0xfc, 0x1f, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x3f, 0xfe, 0x00, 0x00, 0x03, 0xfe, 0x00, 0x00,
 0xfc, 0x3f, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x3f, 0xfe, 0x00, 0x00, 0x07, 0xfc, 0x00, 0x00, 0xfc,
 0x3f, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x37, 0xfc, 0x00, 0x00, 0x0f, 0xf8, 0x00, 0x00, 0xfc, 0x3f,
 0xff, 0xc0, 0x00, 0x00, 0x00, 0x1f, 0xfc, 0x00, 0x00, 0x0f, 0xfc, 0x00, 0x01, 0xfc, 0x3f, 0x9f,
 0xc0, 0x00, 0x00, 0x00, 0x1f, 0xfc, 0x00, 0x00, 0x1f, 0xfe, 0x00, 0x01, 0xfc, 0x1f, 0x1f, 0xc0,
 0x00, 0x00, 0x00, 0x1f, 0xfc, 0x00, 0x00, 0x3f, 0xfe, 0x00, 0x03, 0xfc, 0x0c, 0x1f, 0xc0, 0x00,
 0x00, 0x00, 0x0f, 0xf8, 0x00, 0x00, 0x7f, 0xfe, 0x00, 0x07, 0xf8, 0x00, 0x1f, 0xc0, 0x00, 0x00,
 0x00, 0x0f, 0xf8, 0x00, 0x00, 0x7f, 0xfe, 0x00, 0x0f, 0xf8, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00,
 0x0f, 0xf8, 0x00, 0x00, 0xff, 0xfe, 0x07, 0x3f, 0xf0, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x0f,

0xf8, 0x00, 0x01, 0xff, 0xfe, 0x7f, 0xff, 0xe0, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x0d, 0xf8,
 0x00, 0x01, 0xfe, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x05, 0xf8, 0x00,
 0x03, 0xfc, 0xff, 0xff, 0xff, 0x80, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x05, 0xf8, 0x00, 0x07,
 0xf8, 0xff, 0xff, 0xff, 0x00, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x05, 0xf0, 0x00, 0x0f, 0xf8,
 0xff, 0xff, 0xff, 0x80, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x05, 0xf0, 0x00, 0x0f, 0xf0, 0xff,
 0xff, 0xff, 0x80, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x05, 0xf0, 0x00, 0x1f, 0xe0, 0xfe, 0x7f,
 0xff, 0xc0, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x05, 0xf0, 0x00, 0x3f, 0xe0, 0x7e, 0x00, 0x3f,
 0xc0, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x05, 0xf0, 0x00, 0xff, 0xc0, 0x7e, 0x00, 0x1f, 0xe0,
 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x05, 0xf0, 0x00, 0xff, 0x80, 0x7e, 0x00, 0x0f, 0xe0, 0x00,
 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x05, 0xf0, 0x00, 0xff, 0x00, 0xfe, 0x00, 0x0f, 0xe0, 0x00, 0x1f,
 0xc0, 0x00, 0x00, 0x00, 0x05, 0xf0, 0x00, 0xfe, 0x00, 0xfe, 0x00, 0x07, 0xe0, 0x00, 0x1f, 0xc0,
 0x00, 0x00, 0x00, 0x05, 0xf8, 0x00, 0xfe, 0x00, 0xfe, 0x00, 0x07, 0xe0, 0x00, 0x1f, 0xc0, 0x00,
 0x00, 0x00, 0x05, 0xf8, 0x00, 0xfc, 0x00, 0xfe, 0x00, 0x07, 0xf0, 0x00, 0x1f, 0xc0, 0x00, 0x00,
 0x00, 0x0d, 0xf8, 0x00, 0x30, 0x00, 0xfe, 0x00, 0x07, 0xf0, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00,
 0x0f, 0xf8, 0x00, 0x00, 0x00, 0xfe, 0x00, 0x07, 0xe0, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x0f,
 0xf8, 0x00, 0x00, 0x00, 0xfe, 0x00, 0x0f, 0xe0, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x0f, 0xf8,
 0x00, 0x00, 0x00, 0xfe, 0x00, 0x0f, 0xe0, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x0f, 0xf8, 0x00,
 0x00, 0x00, 0xfe, 0x00, 0x1f, 0xe0, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x1f, 0xfc, 0x00, 0x00,
 0x00, 0xfe, 0x00, 0x3f, 0xc0, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x1f, 0xfc, 0x00, 0x00, 0x00,
 0xfe, 0x00, 0x7f, 0xc0, 0x00, 0x1f, 0xe0, 0x00, 0x00, 0x00, 0x1f, 0xfc, 0x00, 0x00, 0x00, 0xff,
 0xff, 0xff, 0x80, 0x00, 0x1f, 0xe0, 0x00, 0x00, 0x00, 0x1f, 0xfc, 0x00, 0x00, 0x00, 0xff, 0xff,
 0xff, 0x80, 0x00, 0x1f, 0xe0, 0x00, 0x00, 0x00, 0x1f, 0xfe, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff,
 0x00, 0x00, 0x0f, 0xc0, 0x00, 0x00, 0x00, 0x3f, 0xfe, 0x00, 0x00, 0x00, 0xff, 0xff, 0xfe, 0x00,
 0x00, 0x07, 0xc0, 0x00, 0x00, 0x00, 0x3f, 0xfe, 0x00, 0x00, 0x00, 0x7f, 0xff, 0xf8, 0x00, 0x00,
 0x03, 0x00, 0x00, 0x00, 0x00, 0x7f, 0xfe, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xe0, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x7f, 0xff, 0x00, 0x00, 0x00, 0x03, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x7f, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01,
 0xff, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff,
 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xff, 0xff,
 0xe0, 0x00, 0x0f, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x07, 0xff, 0xff, 0xe0,
 0x00, 0x1f, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x07, 0xff, 0xff, 0xf0, 0x00,
 0x3f, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xf0, 0x00, 0x3c,
 0x38, 0x08, 0x04, 0x00, 0x01, 0x80, 0xe7, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xf8, 0x00, 0x3c, 0x00,
 0x7f, 0x0e, 0x1c, 0x7f, 0xc1, 0xff, 0x00, 0x00, 0x1f, 0xff, 0xff, 0xfc, 0x00, 0x3e, 0x00, 0xff,
 0x8e, 0x1c, 0x7f, 0xe3, 0xff, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xfc, 0x00, 0x1f, 0xc0, 0xe3, 0x8e,
 0x1c, 0x79, 0xe3, 0x8f, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xfe, 0x00, 0x0f, 0xf1, 0xc3, 0xce, 0x1c,
 0x70, 0xe7, 0x87, 0x00, 0x00, 0x7f, 0xff, 0xff, 0xff, 0x00, 0x01, 0xf9, 0xc1, 0xce, 0x1c, 0x70,

```

0xe7, 0x87, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x79, 0xc1, 0xce, 0x1c, 0x70, 0xe7,
0x87, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x39, 0xc1, 0xcf, 0x1c, 0x70, 0xe3, 0x87,
0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x3c, 0x78, 0xe3, 0x8f, 0xbc, 0x70, 0xe3, 0xdf, 0x00,
0x03, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x3f, 0xf8, 0xff, 0x87, 0xfc, 0x70, 0xe3, 0xff, 0x00, 0x07,
0xff, 0xff, 0xff, 0xff, 0xf0, 0x0f, 0xe0, 0x7f, 0x07, 0xfc, 0x70, 0xe1, 0xf7, 0x00, 0x0f, 0xff,
0xff, 0xff, 0xff, 0xf8, 0x01, 0x00, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xff,
0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xff,
0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0x00,
0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00,
0x00, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x01,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x1f, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x01, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff
};

```

```

void setup( ) {
    oled.begin( 0x3c );
    oled.clearDisplay( );
    oled.drawBitmap( 0, 0, myBTSound, 25, 25, WHITE );
    oled.display( );
    delay( 1000 );
}

```

```

void loop( ) {
}

```

ឧទាហរណ៍ ចូរប្រើ OLED គូសឱ្យចេញរូបរលកស៊ីញ៉ាល់ ហើយអាចប្តូរប្រេកង់បានតាម Serial Monitor

```

#include <Adafruit_GFX.h>
#include <gfont.h>

```

```

#include <Adafruit_SSD1306.h>
Adafruit_SSD1306 oled( LED_BUILTIN );
float A = 2;
float F = 3;
float phi = 4;
void setup( ) {
    Serial.begin( 9600 );
    oled.begin( 0x3c );
    oled.clearDisplay( );
    oled.display( );
    drawwave( A, F, phi );
}

void loop( ) {
    if( Serial.available( ) > 0 ){
        String txt = Serial.readString( );
        if( txt.startsWith( "F" ) ){
            F = txt.substring( 1 ).toFloat( );
            drawwave( A, F, phi );
        }
    }
}

void drawwave( float A, float F, float phi ){
    oled.clearDisplay( );
    for ( int t = 0; t < 128; t++){
        int y = A * sin( 2 * 3.14 * F * t + phi );
        oled.drawPixel( t, 35 + y, WHITE );
    }
    oled.display( );
}

```

គ. ការប្រើ Real Time ជាមួយ OLED

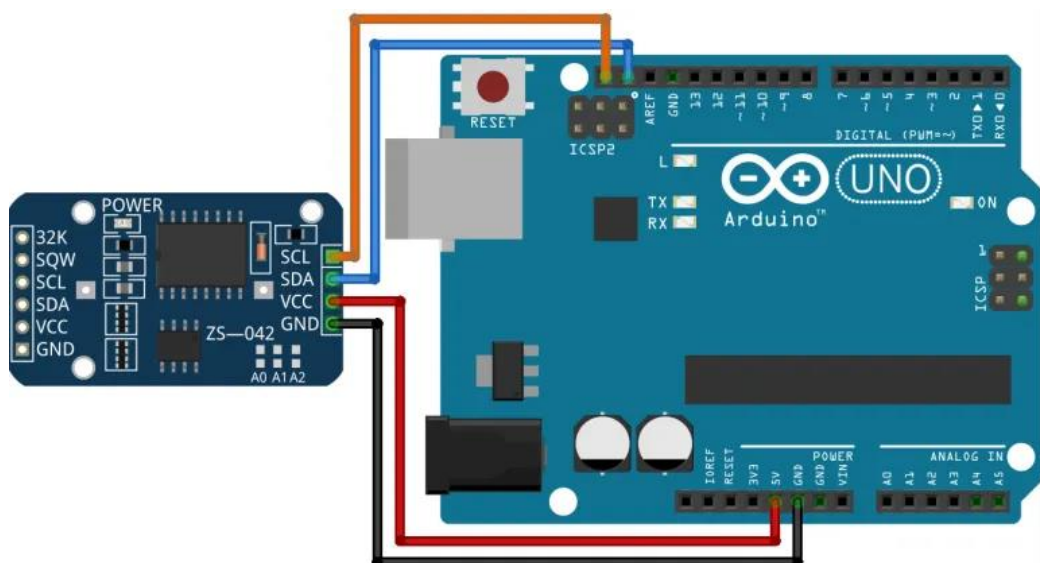


រូបទី៦.១៤ Real Time Module

យើងអាចប្រើប្រាស់ Real Time Module ដែលយើងអាចទាញយកទិន្នន័យម៉ោង ថ្ងៃ ខែ ពី Real Time បាន លក្ខណៈពិសេសរបស់ Real Time គឺមាន Battery Backup ដែលអាចចងចាំទិន្នន័យពេលវេលាបានល្អ បើទោះជាដាច់ភ្លើងក៏ដោយ ។ ដើម្បីអាចប្រើវាបាន យើងត្រូវមាន Library មួយដែលមានឈ្មោះថា DS1307 ពីគេហទំព័រ៖

http://www.rinkydinkelectronics.com/no_deep_linking.php?file=DS1307.zip

ឧទាហរណ៍១ ការប្រើ Serial Monitor ជាមួយ Real Time ដោយទាញម៉ោង ថ្ងៃ ខែ ពី Real Time មកបង្ហាញលើ Serial Monitor។



រូបទី៦.១៥ ការតភ្ជាប់ Real Time ជាមួយ Arduino

```

#include <Wire.h>
#include <RTClib.h>
RTC_DS3231 myRTC;

// ប្រសិនបើប្រើ DS3231 ឬ RTC_DS3231 (យកតាមឧទាហរណ៍នេះ)
// តែបើប្រើ DS1307 ត្រូវប្តូរដាក់ RTC_DS1307
// តែបើប្រើ PCF8523 ត្រូវប្តូរដាក់ RTC_PCF8523
// ហើយកំណត់ឈ្មោះថា myRTC

void setup( )
{
    Serial.begin( 9600 );
    Wire.begin( );
}

void loop( )
{
    DateTime nowTime = myRTC.now( );
    Serial.print( "Now date/time :: " );
    Serial.print( nowTime.hour( ) );
    Serial.print( ":" );
    Serial.print( nowTime.minute( ) );
    Serial.print( ":" );
    Serial.print( nowTime.second( ) );
    Serial.print( " :: " );
    Serial.print( nowTime.day( ) );
    Serial.print( "/" );
    Serial.print( nowTime.month( ) );
    Serial.print( "/" );
    Serial.println( nowTime.year( ) );
    delay( 1000 );
}

```

ឧទាហរណ៍២ កំណត់ម៉ោង បិទបើកអំពូល និងបង្ហាញម៉ោងនៅលើ OLED ។

```

#include <Adafruit_GFX.h>
#include <gfont.h>
#include <Adafruit_SSD1306.h>

#include <DS1307.h>
DS1307 myrtc ( SDA,SCL );
Adafruit_SSD1306 myoled( 13 );

void setup( ) {
    myrtc.begin( );
    Serial.begin( 9600 );
    // myrtc.setDate( 27,6,2020 );
    // myrtc.setTime( 16,54,00 );
    // myrtc.setDOW( SATURDAY );
    myoled.begin( 0x3c );
    myoled.setTextSize( 2 );
    myoled.setTextColor( WHITE );
    pinMode( A0,OUTPUT );
}

void loop( ) {
    myoled.setCursor( 0,0 );
    myoled.print( myrtc.getDateStr( ) );
    myoled.setCursor( 0,20 );
    myoled.print( myrtc.getTimeStr( ) );
    myoled.setCursor( 0,45 );
    myoled.print( myrtc.getDOWStr( ) );
    myoled.display( );
    Time t = myrtc.getTime( );
    int h=t.hour;
    int m=t.min;
    if ( ( h==16 ) && ( m==55 ) ){

```

```

        digitalWrite( A0,1 );
    }
    else if ( m>55){
        digitalWrite( A0,0 );
    }
    delay( 1000 );
    myoled.clearDisplay( );
}

```

ឧទាហរណ៍៣ បង្កើតនាឡិកា Analog នៅលើ OLED ។

```

#include <Adafruit_SSD1306.h>
Adafruit_SSD1306 oled( 13 );
int pre_x,pre_y,pre_xm,pre_ym,pre_xh,pre_yh;
int xm,ym,xh,yh;

void setup( ) {
    oled.begin( 0x3c );
    oled.display( );
    oled.clearDisplay( );
    oled.drawCircle( 63,31,31,WHITE );
    for( int i = 0; i<360;i+=30){
        int x1 = 63+25 *cos( i *0.0174 - 3.14/2 );
        int y1 = 31+25 *sin( i *0.0174 - 3.14/2 );
        int x2 = 63+27 *cos( i *0.0174 - 3.14/2 );
        int y2 = 31+27 *sin( i *0.0174 - 3.14/2 );
        oled.drawLine( x1,y1,x2,y2,WHITE );
        oled.display( );
    }
}

void loop( ) {
    for( int h = 0; h<360;h+=6){

```

```

    xh = 63+10 *cos( h *0.0174 - 3.14/2 );
    yh = 31+10 *sin( h *0.0174 - 3.14/2 );
    oled.drawLine( 63,31,pre_xh,pre_yh,BLACK );
    for( int m = 0; m<360;m+=6 ){
        xm = 63+15 *cos( m *0.0174 - 3.14/2 );
        ym = 31+15 *sin( m *0.0174 - 3.14/2 );
        oled.drawLine( 63,31,pre_xm,pre_ym,BLACK );
    }
    for( int i = 0; i<360;i+=6 ){
        int x = 63+20 *cos( i *0.0174 - 3.14/2 );
        int y = 31+20 *sin( i *0.0174 - 3.14/2 );
        oled.drawLine( 63,31,pre_x,pre_y,BLACK );
        oled.drawLine( 63,31,xh,yh,WHITE );
        oled.drawLine( 63,31,xm,ym,WHITE );
        oled.drawLine( 63,31,x,y,WHITE );
        oled.display( );

        pre_x = x;
        pre_y = y;
        pre_xm = xm;
        pre_ym = ym;
        pre_xh = xh;
        pre_yh = yh;
        delay( 1000 );
    }
}
}
}

```

ឧទាហរណ៍៤ បង្កើតនាឡិកា Digital នៅលើ OLED ។

```

#include <Adafruit_SSD1306.h>
#include <DS1307.h>
Adafruit_SSD1306 oled( LED_BUILTIN );
DS1307 rtc( SDA, SCL );
float pha = PI / 2;

```

```

float xc = 80;
float yc = 31;
float r1 = 25;
float r2 = 30;
float rs = 23;
float rm = 18;
float rh = 13;
float t;
float degs, pdegs;
float degm, pdegm;
float deggh, pdeggh;
void setup( ) {
    Serial.begin( 9600 );
    rtc.begin( );
    rtc.halt( false );
    oled.begin( 0x3c );
    oled.clearDisplay( );
    oled.drawCircle( xc, yc, 30, WHITE );
    for ( int i = 0; i < 360; i += 30 ) {
        drawtick( r1, r2, i, 1 );
    }
    oled.display( );
}
void loop( ) {
    unsigned long t1 = millis( );
    String timstr = rtc.getTimeStr( );
    oled.fillRect( 0, 0, 50, 10, 0 );
    oled.setTextSize( 1 );
    oled.setTextColor( 1 );
    oled.setCursor( 0, 0 );
    oled.print( timstr );
    int ind1 = timstr.indexOf( ":" );
    int ind2 = timstr.lastIndexOf( ":" );

```

```

float h = timstr.substring( 0, ind1 ).toFloat( );
float m = timstr.substring( ind1 + 1, ind2 ).toFloat( );
float s = timstr.substring( ind2 + 1 ).toFloat( );
t = h * 3600 + m * 60 + s;
degs = t * 6;

degm = degs * 6 / 360;           //when sec rotate 360deg we got 6deg minute
degh = degs * 30 / 21600;       //when sec rotate 21600degree we got 30deg for hours
oled.fillCircle( xc, yc, r1, 0 );
drawtick( 0, rs, degs, 1 );
drawtick( 0, rm, degm, 1 );
drawtick( 0, rh, degh, 1 );
pdegs = degs;
pdegm = degm;
pdegh = degh;
unsigned long t2 = millis( );
delay( 1000 - ( t2 - t1 ) );
oled.display( );
}

void drawtick( float r_, float R_, float deg, int color ) {
    int xp1 = xc + r_ * cos( -deg * PI / 180 + pha );
    int yp1 = yc - r_ * sin( -deg * PI / 180 + pha );
    int xp2 = xc + R_ * cos( -deg * PI / 180 + pha );
    int yp2 = yc - R_ * sin( -deg * PI / 180 + pha );
    oled.drawLine( xp1, yp1, xp2, yp2, color );
}

```

ឧទាហរណ៍៥ បង្កើតនាឡិកា Analog និង Digital នៅលើ OLED ហើយអាចប្តូរម៉ោង ថ្ងៃ ខែបានតាមរយៈ Serial Monitor។

```

#include <Adafruit_SSD1306.h>
#include <DS1307.h>
Adafruit_SSD1306 oled( LED_BUILTIN );
DS1307 rtc( SDA, SCL );

```

```

float pha = PI / 2;
float xc = 80;
float yc = 31;
float r1 = 25;
float r2 = 30;
float rs = 23;
float rm = 18;
float rh = 13;
float t;
float degs, pdegs;
float degm, pdegm;
float deggh, pdeggh;
void setup( ) {
  Serial.begin( 9600 );
  rtc.begin( );
  rtc.halt( false );
  oled.begin( 0x3c );
  oled.clearDisplay( );
  oled.drawCircle( xc, yc, 30, WHITE );
  for ( int i = 0; i < 360; i += 30 ) {
    drawtick( r1, r2, i, 1 );
  }
  oled.display( );
}
void loop( ) {
  if ( Serial.available( ) > 0 ){
    String data = Serial.readString( );
    Serial.println( data );
    if ( data.startsWith( "dat" ) ){
      int d=data.substring( 3,5 ).toInt( );
      int m=data.substring( 6,8 ).toInt( );
      int y=data.substring( 9,13 ).toInt( );
      rtc.setDate( d,m,y );
    }
  }
}

```

```

    }
    else if ( data.startsWith( "tim" ) ){
        int h=data.substring( 3,5 ).toInt( );
        int mn=data.substring( 6,8 ).toInt( );
        int s=data.substring( 9,11 ).toInt( );
        rtc.setTime( h,mn,s);
    }
    else if ( data.startsWith( "dow" ) ){
        int D=data.substring( 3,4 ).toInt( );
        rtc.setDOW( D );
    }
}

unsigned long t1 = millis( );
String timstr = rtc.getTimeStr( );
oled.fillRect( 0, 10, 50, 10, 0 );
oled.setTextSize( 1 );
oled.setTextColor( 1 );
oled.setCursor( 0,10 );
oled.print( timstr );

oled.setCursor( 0, 0 );
oled.print( rtc.getDateStr( ) );

int ind1 = timstr.indexOf( ":" );
int ind2 = timstr.lastIndexOf( ":" );
float h = timstr.substring( 0, ind1 ).toFloat( );
float m = timstr.substring( ind1 + 1, ind2 ).toFloat( );
float s = timstr.substring( ind2 + 1 ).toFloat( );
t = h * 3600 + m * 60 + s;
degs = t * 6;
degm = degs * 6 / 360; //when sec rotate 360deg we got 6deg minute
deggh = degs * 30 / 21600; //when sec rotate 21600degree we got 30deg for hours
oled.fillCircle( xc, yc, r1, 0 );

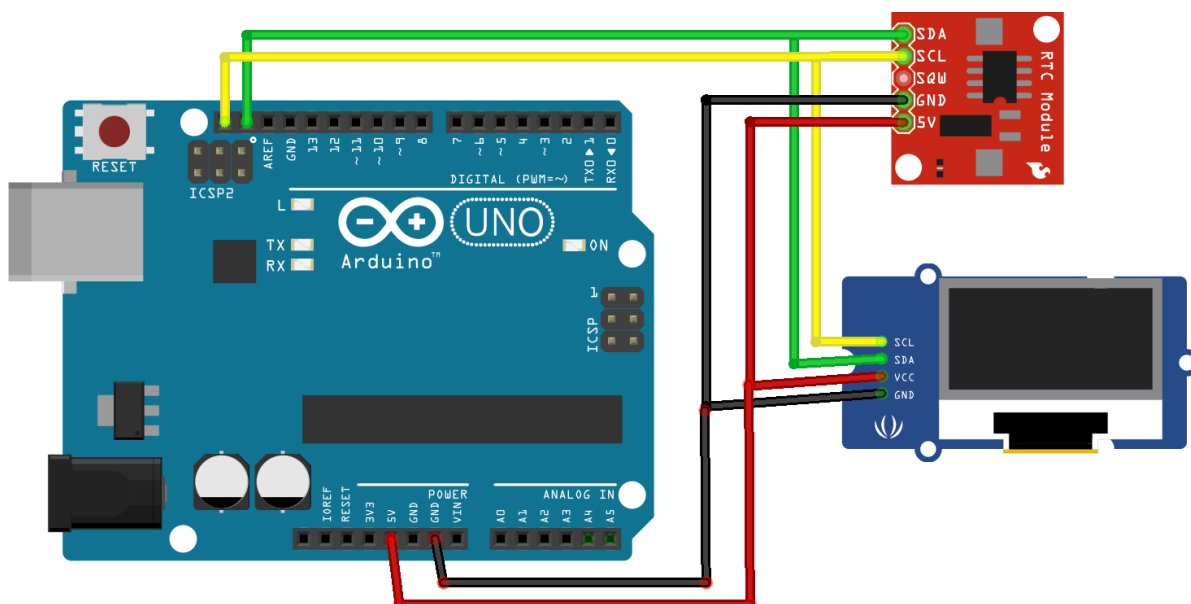
```

```

drawtick( 0, rs, degs, 1);
drawtick( 0, rm, degm, 1);
drawtick( 0, rh, deggh, 1);
pdegs = degs;
pdegm = degm;
pdeggh = deggh;
unsigned long t2 = millis( );
delay( 1000 - ( t2 - t1 ) );
oled.display( );
}

void drawtick( float r_, float R_, float deg, int color ) {
  int xp1 = xc + r_ * cos( -deg * PI / 180 + pha );
  int yp1 = yc - r_ * sin( -deg * PI / 180 + pha );
  int xp2 = xc + R_ * cos( -deg * PI / 180 + pha );
  int yp2 = yc - R_ * sin( -deg * PI / 180 + pha );
  oled.drawLine( xp1, yp1, xp2, yp2, color );
}

```

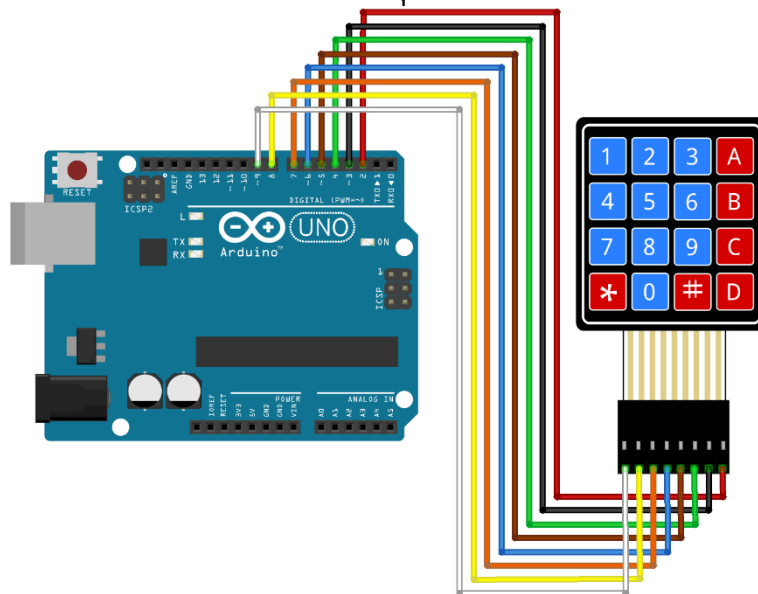


រូបទី៦.១៦ ការតភ្ជាប់ឧទាហរណ៍ទី៣ Real Time, oled និង board Arduino

មេរៀនទី៧ EEPROM និង SD Card

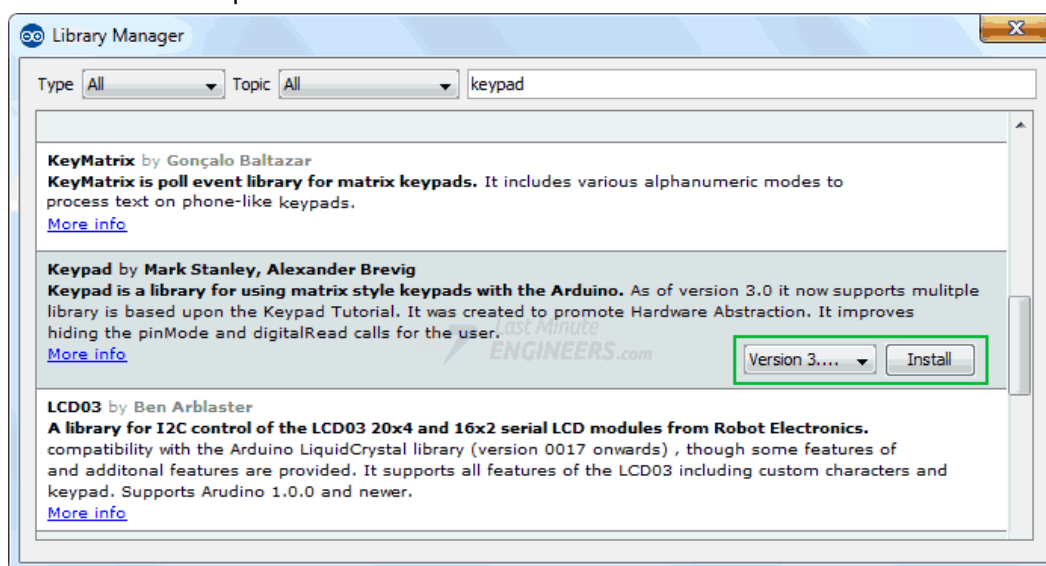
១. Keypad

Keypad គឺជា keyboard ដែលមានតែលេខនិងអក្សរមួយចំនួនតូចប៉ុណ្ណោះ សម្រាប់ដាក់បញ្ចូលលេខឬអក្សរទៅក្នុង Arduino ហើយមាន ច្រើនប្រភេទឱ្យយើងប្រើតាមតម្រូវការ ។ ដើម្បីអាចប្រើ Keypad ចាំបាច់យើងត្រូវទាញយក Library មកប្រើពីគេហទំព័រ <https://github.com/Chris--A/Keypad> ។ សម្រាប់ការភ្ជាប់ Keypad ចូល board Arduino យើងអាចតភ្ជាប់ក្នុងរូបទី៧.១។



រូបទី៧.១ ភ្ជាប់ keypad ចូល board Arduino

Library keypad យើងអាចទាញយកមកប្រើបាន ដោយចុចលើពាក្យ **Sketch > Include Library > Manage Libraries...** បន្ទាប់មកសរសេរពាក្យ keypad ក្នុង search ហើយធ្វើការដំឡើង file មួយដែលមានឈ្មោះថា keypad ដូចក្នុងរូបទី៧.២ ។



រូបទី៧.២ ការដំឡើង Library keypad

ក្នុងការសរសេរកូដប្រើ Keypad យើងមានពាក្យបញ្ជាមួយចំនួនត្រូវដឹងដូចតទៅនេះ៖

- **Constructor** ប្រើសម្រាប់ប្រកាស Library keypad ដែលមានទម្រង់ដូចខាងក្រោម៖

ទម្រង់ 4x3

Keypad.h

```
const byte ROWS = 4;
```

```
const byte COLS = 3;
```

```
char keys[ROWS][COLS] = {
```

```
    {'1', '2', '3'},
```

```
    {'4', '5', '6'},
```

```
    {'7', '8', '9'},
```

```
    {'*', '0', '#'}

```

```
};
```

```
byte rowPins[ROWS] = {13, 12, 11, 10}; //connect to the row pinouts of the keypad
```

```
byte colPins[COLS] = {9, 8, 7}; //connect to the column pinouts of the keypad
```

ទម្រង់ 4x4

Keypad.h

```
const byte ROWS = 4;
```

```
const byte COLS = 4;
```

```
char keys[ROWS][COLS] = {
```

```
    {'1', '2', '3', 'A'},
```

```
    {'4', '5', '6', 'B'},
```

```
    {'7', '8', '9', 'C'},
```

```
    {'*', '0', '#', 'D'}

```

```
};
```

```
byte rowPins[ROWS] = {13, 12, 11, 10}; //connect to the row pinouts of the keypad
```

```
byte colPins[COLS] = {9, 8, 7, 6}; //connect to the column pinouts of the keypad
```

- **waitForKey()** ជាទម្រង់ដែលរង់ចាំរហូតទាល់តែគេចុចប៊ូតុង។ នៅពេលវាដើរ រាល់កម្មវិធីទាំងអស់ត្រូវ block រហូតដល់មាន keyboard ណាមួយត្រូវបានចុច។
- **getKey()** សម្រាប់ទាញយកតម្លៃពី key ណាមួយដែលយើងចុច។

- **getState()** សម្រាប់មើលពីស្ថានភាពរបស់ key ថាគឺជាស្ថិតក្នុងសភាពបែបណា (IDLE, PRESSED, RELEASED, HOLD) ។
- **setHoldTime()** សម្រាប់កំណត់រយៈពេលគិតជាមីលីវិនាទី ដែលយើងត្រូវចុចប៊ូតុងឱ្យជាប់រហូតដល់រយៈពេលនោះទើបវាដំណើរការបាន ។ **serHoldTime(unsigned int time)**
- **setDebounceTime()** កំណត់រយៈពេលគិតជាមីលីវិនាទីដែលសម្រាប់ការពារពី Debouncer/Noise ។ **setDebounceTime(unsigned int time)**
- **addEventListener()** នៅពេលដែល key ត្រូវបានចុច Function ចាប់ដំណើរការ **addEventListener(Keypad Event)**

ឧទាហរណ៍១ ចុចលើ keypad ហើយបង្ហាញលើ Serial Monitor

```
#include <Keypad.h>

const byte row=4;
const byte col=4;
char key [row][col]={
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};

byte rowpin[row]={9,8,7,6};
byte colpin[col]={5,4,3,2};
Keypad mykeypad=Keypad( makeKeymap( key ),rowpin,colpin,row,col );
void setup( ) {
    Serial.begin( 9600 );
}

void loop( ) {
    char mykey = mykeypad.getKey( );
    if ( mykey>0 ){
        Serial.println( mykey );
    }
}
```

ឧទាហរណ៍២ ចុចលើ keypad ហើយបង្ហាញលើ LCD

```

#include <Wire.h>
#include <LiquidCrystal.h>
#include <Keypad.h>
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'} };
byte rowPins[ROWS] = {13, 12, 11, 10}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {9, 8, 7, 6}; //connect to the column pinouts of the keypad
Keypad keypad = Keypad( makeKeymap( keys ), rowPins, colPins, ROWS, COLS );
LiquidCrystal lcd( 0,1,2,3,4,5 );
void setup( ) {
    lcd.begin( 20,4 );
    lcd.home( );
    lcd.setCursor( 0,0 );
    lcd.print( "   Keypad Test   ");
    lcd.setCursor( 0,3 );
    lcd.print( "Key       =       ");
    lcd.blink( );
    lcd.setCursor( 19,3 );
}
void loop( ) {
    {char key;
    key = keypad.getKey( );
    if ( key != NO_KEY ) {
        lcd.print( key );
        lcd.setCursor( 19,3 );
    }
    }
}

```

ដើម្បីឱ្យដឹងថាយើងបានចុច keypad ពិត ឬអត់ យើងអាចបន្ថែម Buzzer ដែលវាមានសម្លេងប្រាប់ថាបានចុច keypad ពិតប្រាកដ។

ឧទាហរណ៍៣ ចុចលើ keypad ហើយបង្ហាញលើ LCD និងមានសម្លេង(Buzzer ចូលជើងទី A1)។

```
#include <Wire.h>
#include <LiquidCrystal.h>
#include <Keypad.h>
int Buzzer = A1;
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {13, 12, 11, 10}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {9, 8, 7, 6}; //connect to the column pinouts of the keypad

Keypad keypad = Keypad( makeKeymap( keys ), rowPins, colPins, ROWS, COLS );
LiquidCrystal lcd( 0,1,2,3,4,5 );
void setup( ){
  lcd.begin( 20,4 );
  lcd.home( );
  lcd.setCursor( 0,0 );
  lcd.print( "  Keypad Test  ");
  lcd.setCursor( 0,3 );
  lcd.print( "Key    =    ");
  pinMode( Buzzer,OUTPUT );
  digitalWrite( Buzzer,LOW );
  lcd.blink( );
  lcd.setCursor( 19,3 );
}
```

```

void loop( ) {
    char key;
    key = keypad.getKey( );
    if ( key != NO_KEY )
    {
        lcd.print( key );
        lcd.setCursor( 19,3);
        Beep( );
    }
}

void Beep( )
{
    digitalWrite( Buzzer,HIGH );delay( 50 );
    digitalWrite( Buzzer,LOW );
}

```

ឧទាហរណ៍៤ ចូរអ្នកសរសេរកូដប្រព័ន្ធទ្វារសុវត្ថិភាពដែលមាន OLED មួយសម្រាប់បង្ហាញពាក្យថា Lock or Unlock និង Password ដែលយើងបញ្ចូលទៅតាមរយៈ keypad។ នៅពេលដែល Password ដាក់ទៅខុស វានឹងបញ្ចេញសម្លេងតាម Buzzer ។ គួរបញ្ជាក់ថារាល់ពេលបញ្ចូល Password រួចត្រូវចុច សញ្ញា # ដើម្បី Activate Password ។

```

#include <Adafruit_SSD1306.h>
#include <Adafruit_GFX.h>
#include <gfont.h>
#include <Key.h>
#include <Keypad.h>
char keys[4][4]={
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}};
byte row[] ={9,8,7,6};
byte col[] ={5,4,3,2};

```

```
Keypad pad=Keypad( makeKeymap( keys ),row,col,4,4 );
Adafruit_SSD1306 oled( LED_BUILTIN );
String password ="1234";
String input="";          // "" មានន័យថាលុបចាស់ៗពី OLED
```

```
void setup( ){
    Serial.begin( 9600 );
    oled.begin( 0x3c );
    oled.clearDisplay( );
    oled.setTextSize( 2 );
    oled.setTextColor( WHITE );
    oled.setCursor( 0,0 );
    oled.print( "LOCK" );
    oled.display( );
    pinMode( 13,OUTPUT );
    pad.setDebounceTime( 300 );
}

void loop( ) {
    char keycode = pad.getKey( );
    if( keycode ){
        if( keycode !='#' ){
            input = input+String( keycode );
            oled.setCursor( 0,20 );
            oled.print( input );
            oled.display( );
        } else {
            if( input == password ){
                oled.clearDisplay( );
                oled.setCursor( 0,0 );
                oled.print( "UNLOCK" );
                oled.display( );
                delay( 10000 );
            } else {
```

```

oled.clearDisplay( );
oled.setCursor( 0,0);
oled.print( "Wrong" );
oled.display( );
delay( 3000 );
}
input = "";
oled.clearDisplay( );
oled.setCursor( 0,0);
oled.print( "LOCK" );
oled.display( );
}
}
}

```

២. EEPROM

EEPROM ត្រូវបានប្រើសម្រាប់ការស្តុកទិន្នន័យរយៈពេលយូរដែលអាចបានរហូតដល់ទៅ ២០ឆ្នាំ។ ជាការពិតណាស់ អញ្ញាតនៅក្នុង Arduino មិនអាចផ្ទុកទិន្នន័យបានទេនៅពេលវាដាច់ភ្លើងហើយការដាក់ទិន្នន័យទៅក្នុង EEPROM ជាការចាំបាច់ដើម្បីរក្សាវា ទុកប្រើពេលមានភ្លើងវិញ។

ក. ពាក្យបញ្ជាដែលត្រូវប្រើជាមួយ EEPROM

- **read()** វាជា function ប្រើសម្រាប់អានទិន្នន័យយកទិន្នន័យ ១byte ពី Address ណាមួយរបស់ EEPROM
ទម្រង់
EEPROM.read(address)
- **write()** ជា function ប្រើសម្រាប់ដាក់បញ្ចូលទិន្នន័យ 1byte ពី Address ណាមួយរបស់ EEPROM
ទម្រង់
EEPROM.read(address, value)
- **update()** ជា function ប្រើសម្រាប់ដាក់បញ្ចូលទិន្នន័យ 1byte ពី Address ណាមួយរបស់ EEPROM តែការដាក់នេះគឺធ្វើការបានលុះត្រាតែមានទិន្នន័យដែលដាក់ខុសពីទិន្នន័យដែលមានស្រាប់នៅក្នុង Address នោះ។

ទម្រង់

EEPROM.update (address, value)

▪ **get()**

សម្រាប់ទាញយកទិន្នន័យជាប្រភេទណាមួយក៏បានរួចបញ្ចូលទាំងជាប្រភេទ Float

ទម្រង់

EEPROM.put(address,data)

▪ **put()** សម្រាប់ដាក់ទិន្នន័យជាប្រភេទណាមួយក៏បានបញ្ចូលទាំងជាប្រភេទ Float**ទម្រង់**

EEPROM.put(address, value)

▪ **EEPROM[]** សម្រាប់ដាក់ ឬទាញយកទិន្នន័យពី Address ជាក់លាក់ណាមួយ**ទម្រង់**

EEPROM[address]

ឧទាហរណ៍១ ប្រើ EEPROM ដើម្បីដាក់ និងទាញយក 1 byte

```
#include <EEPROM.h>

void setup( )
{
    Serial.begin( 115200 );
    int j = EEPROM.read( 7 );
    Serial.println( j );
}

void loop( )
{
    if( Serial.available( ) > 0 ){
        String txt = Serial.readString( );
        int t = txt.toInt( );
        EEPROM.write( 7,t );
    }
}
```

ឧទាហរណ៍១ ប្រើ EEPROM ដើម្បីដាក់ និងទាញយក Data object

```
#include<EEPROM.h>

char x[20];

void setup( ){
    Serial.begin( 115200 );
    EEPROM.get( 15,x );
    Serial.println( String( x ) );
}

void loop( ){
    if ( Serial.available( )>0 ){
        String txt = Serial.readString( );
        strcpy( x,txt.c_str( ) );
        EEPROM.put( 15,x );
    }
}
```

ឧទាហរណ៍២ ចូរធ្វើឱ្យដូចអនុវត្តន៍ទ្វារសុវត្ថិភាពខាងលើអ្នកត្រូវរក្សាទុកនិងយក Password របស់អ្នកពី EEPROM ដើម្បីដាក់ Password ថ្មីអ្នកត្រូវចុច * ដើម្បី Save New Password ។

```
#include<EEPROM.h>
#include<Adafruit_SSD1306.h>
#include<Key.h>
#include<Keypad.h>

char keys[4][4]={
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};

byte row[] ={9,8,7,6};
byte col[] ={5,4,3,2};

Keypad pad=Keypad( makeKeymap( keys ),row,col,4,4 );
Adafruit_SSD1306 oled( LED_BUILTIN );
String password ;
```

```

String input="";
char x[10];
void setup( ) {
    Serial.begin( 9600 );
    oled.begin( 0x3c );
    oled.clearDisplay( );
    oled.setTextSize( 2 );
    oled.setTextColor( WHITE );
    oled.setCursor( 0,0 );
    oled.print( "LOCK" );
    oled.display( );
    pinMode( 12,OUTPUT );
    pad.setDebounceTime( 300 );
    EEPROM.get( 0,x );
    password = String( x );
}
void loop( ) {
    char keycode=pad.getKey( );
    if( keycode ){
        if( keycode !='#' ){
            if( keycode !='*' ){
                input = input+String( keycode );
                oled.setCursor( 0,20 );
                oled.print( input );
                oled.display( );
            } else {
                strcpy( x, input.c_str( ) );
                EEPROM.put( 0,x );
                password = input;
                oled.clearDisplay( );
                oled.setCursor( 0,0 );
                oled.print( "Saved" );
                oled.display( );
            }
        }
    }
}

```

```

        delay( 3000 );
        input="";
        oled.clearDisplay( );
        oled.setCursor( 0,0 );
        oled.print( "LOCK" );
        oled.display( );
    }
} else {
    if( input==password ){
        oled.clearDisplay( );
        oled.setCursor( 0,0 );
        oled.print( "UNLOCK" );
        oled.display( );
        delay( 10000 );
    }else{
        oled.clearDisplay( );
        oled.setCursor( 0,0 );
        oled.print( "Wrong" );
        oled.display( );
        delay( 3000 );
    }
    input="";
    oled.clearDisplay( );
    oled.setCursor( 0,0 );
    oled.print( "LOCK" );
    oled.display( );
}
}
}

```

៣. SD Card

SD Library ត្រូវបានប្រើសម្រាប់ការទាញយក ឬដាក់ទិន្នន័យចូលទៅកាន់ SD Card memory ។
Library នេះអាចប្រើបានទាំង FAT16 និង FAT32។



រូបទី៧.៣ sd card

- **begin()**

សម្រាប់ផ្ដើមដំណើរការ Library SD។

Syntax:

SD.begin()

SD.begin(cspin)

- **exists()**

សម្រាប់តេស្តមើលថាតើមាន File Name នៅក្នុង SD នោះទេ។

Syntax:

SD.exists(filename)

- **mkdir()**

សម្រាប់បង្កើត directory ក្នុង SD card។ យើងអាចប្រើសញ្ញា % ដើម្បីដាក់ Folder ក្នុង Folder។

Syntax:

SD.mkdir(dirname)

- **open()**

បើក file ពីក្នុង SD Card ដើម្បី Write or Read ។

Syntax:

SD.open(filepath)

SD.open(filepath,mode)

Parameter:

filepath : file name or file name ជាមួយនិង Directory របស់វា

mode : FILE_READ or FILE_WRITE

- **remove()**

សម្រាប់ file នៅក្នុង SD memory

Syntax:

SD.remove(filename)

- **rmdir()**

សម្រាប់លុប Directory នៅក្នុង SD memory

Syntax:

SD.rmdir(dirname)

- **read()**

អានយកទិន្នន័យពី file

Syntax:

SD.read(dirname)

- **name()**

អានយកឈ្មោះរបស់ file

Syntax:

SD.name()

- **available()**

ពិនិត្យមើលថាមានប៉ុន្មាន Bytes នៅក្នុង file

Syntax:

SD.available()

Return:

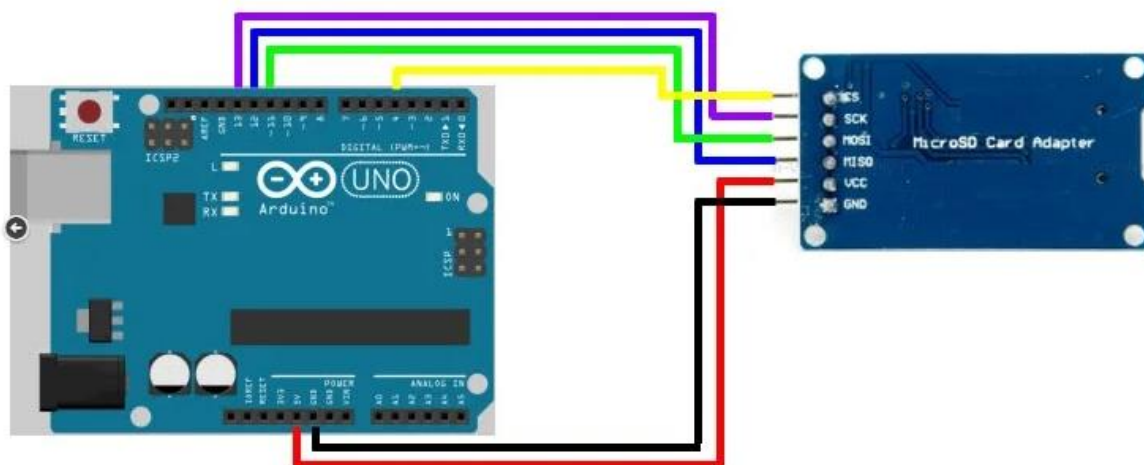
Number of bytes in file

- **close()**

បិទរួច Save File

Syntax:

SD.close()



រូបទី៧.៤ SD card សៀគ្វីសម្រាប់ភ្ជាប់ SD adapter ទៅកាន់ Arduino Uno

ឧទាហរណ៍ ការតេស្ត Arduino ធ្វើការបង្កើត Directory នៅក្នុង SD Card

```
#include<SD.h>
```

```
void setup( ){
```

```
    Serial.begin( 9600 );
```

```
    if ( !SD.begin( 4 ) ){
```

```
        Serial.println( "initialization failed!" );
```

```
        while ( 1 );
```

```
    } else
```

```
    {
```

```
        Serial.println( "initialization OK!" );
```

```
        delay( 2000 );
```

```
        if( SD.exists( "File/a/b" ) == false )
```

```
        {
```

```
            SD.mkdir( "File/a/b" );
```

```
        }
```

```
    }
```

```
}
```

```
void loop( ){
```

```
}
```

Arduino Library មាន Function ដែលអនុញ្ញាតឱ្យអ្នកអាចធ្វើការផ្ទុក ឬអានយកទិន្នន័យរបស់ Memory Card

ទម្រង់ដែលត្រូវប្រើ៖▪ **name()**

សម្រាប់អានយកឈ្មោះរបស់ file ។

Syntax:

file.name()

▪ **available()**

សម្រាប់ពិនិត្យមើលថាមាន ទិន្នន័យ Byte សម្រាប់អានយកនៅក្នុង File ឬអត់

Syntax:

file.available()

▪ **close()**

សម្រាប់បិទ File ហើយ Save

Syntax:

file.close()

▪ **position()**

សម្រាប់ចាប់យកទីតាំងបច្ចុប្បន្នរបស់ File

Syntax:

file.position()

▪ **print()**

ដាក់ទិន្នន័យជាប្រភេទ Byte, Char ,int, long or string ចូល File

Syntax:

file.print(data)

file.print(data,Base)

▪ **println()**

ដាក់ទិន្នន័យជាប្រភេទ Byte, Char ,int, long or string ចូល File បញ្ចប់ដោយ Carriage Return and New fine

Syntax:

file.println(data)

file.println(data,Base)

▪ **size()**

អានយកទំហំ File

Syntax:

file.size()

▪ **read()**

អានយកមួយ Byte នៅក្នុង File

Syntax:

file.read()

▪ **write()**

សម្រាប់ដាក់បញ្ចូលទិន្នន័យមួយ Byte ទៅក្នុង File

Syntax:

file.write()

ឧទាហរណ៍ ការបង្កើត File ដាក់ក្នុង SD Card ហើយធ្វើការដាក់ទិន្នន័យចូល និងអានទិន្នន័យចេញពីក្នុង File នៅក្នុង Memory ។

```
#include<SPI.h>
```

```
#include<SD.h>
```

```
File myFile;
```

```
void setup( ) {
```

```
    Serial.begin( 9600 );
```

```
    if ( !SD.begin( 4 ) ) {
```

```
        Serial.println( "initialization failed" );
```

```
        while ( 1 );
```

```
    }
```

```
    Serial.println( "initialization done." );
```

```
    Serial.println( "Creating test.txt..." );
```

```
    myFile = SD.open( "test.txt", FILE_WRITE );
```

```
    myFile.close( );
```

```
//Check to see if the file exists:
```

```
    if ( SD.exists( "test.txt" ) ) {
```

```
        Serial.println( "test.txt exists." );
```

```
        // open the file .note that only one file can be open at a time,
```

```
        // so you have to close this one before opening another .
```

```
        myFile = SD.open( "test.txt", FILE_WRITE );
```

```
        // if the file opened okay write to it:
```

```

    if ( myFile ) {
        Serial.print( "Writing to test.txt..." );
        myFile.print( "testing 1, 2, 3. " );
        myFile.close( );
        Serial.println( "done. " );
    }
    else {
        // if the file didn't open . print an error
        Serial.println( "error opening test . txt" );
    }

    // re open the file for reading
    myFile = SD.open( "test .txt" );
    if ( myFile ) {
        Serial.println( "test.txt:" );
        // read from the file until there's nothing else in it while ( myFile.available( ) ){
        while ( myFile.available( ) ) {
            Serial.write( myFile.read( ) );
        }
        myFile.close( );
    } else {
        // if the file didn't open , print an error:
        Serial.println( "error opening test.txt" );
    }
} else {
    Serial.println( "test.txt doesn't exist." );
}
}

void loop( ) {
}

```

ក. ការចាត់បទបម្រៀងចេញពី SD card

យើងអាចប្រើប្រាស់ Arduino ជាមួយ SD card សម្រាប់ចាក់បទបម្រៀងដែលយើងបាន Convert ដាក់ក្នុង Memory មកស្តាប់បាន។ ជាដំបូងយើងត្រូវចូលទៅ Download Library នៅក្នុង៖

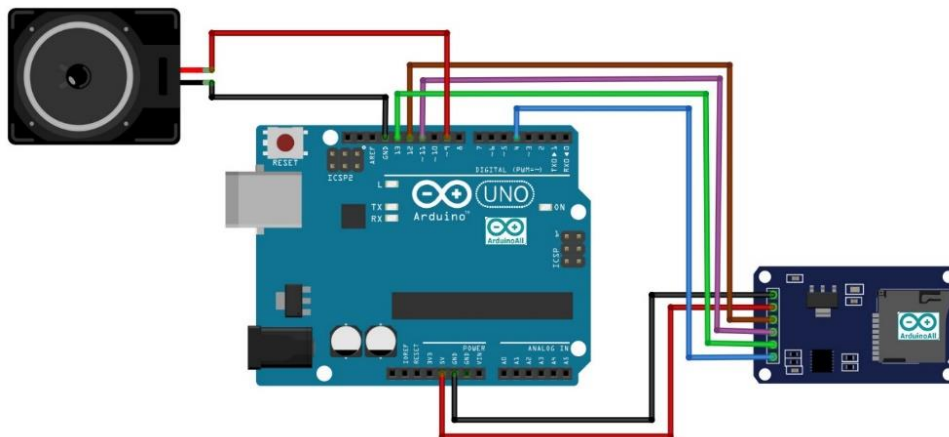
<https://github.com/TMRh20/TMRpcm/archive/master.zip>

បន្ទាប់មកទៀតធ្វើការរៀបចំវាឱ្យក្លាយជាបទចម្រៀងដែលត្រឹមត្រូវសម្រាប់ Arduino Play បានគឺ

- ចម្រៀងត្រូវតែជាប្រភេទ File.wave
- 8 bits
- កំណត់ Change sampling rate: (Hz) = 16000
- 8-32KHz sample rate
- Mono sound

ដើម្បីទទួលបាន File ដូចនេះយើងត្រូវចូលទៅធ្វើការ Convert វាជាមុនសិនជាមួយ Website ខាងក្រោមដែលមានភាពងាយស្រួលក្នុងការ Convert Sound ទៅជា File ប្រភេទ wave។

<https://audio.online-convert.com/convert-to-wave> រួចហើយធ្វើការ Function SD card with FAT32 ហើយចម្លងវាដាក់ក្នុង SD Card ហើយគួរតែកែឈ្មោះដូរឈ្មោះរបស់វាក្នុងករណីដែលឈ្មោះដើមវែង ឬពិបាកចងចាំ ។ រួចហើយអ្នកត្រូវតែសៀវភៅដូចក្នុងរូបទី៧.៥៖



រូបទី៧.៥ សៀវភៅសម្រាប់តភ្ជាប់ SD card និង Speaker ទៅកាន់ Arduino Uno

ឧទាហរណ៍

```
#include<SD.h>
#include<SD.h>
#include<TMRpcm.h>
#include<SPI.h>
#define SD_ChipSelectPin 4
TMRpcm tmrpcm;
void setup( ){
    pinMode( 10, OUTPUT );
    tmrpcm.speakerPin = 9; // 5,6,11 or 4,6 on Mega, 9 on
    if( !SD.begin( SD_ChipSelectPin ) ) {
```

```

digitalWrite( 10, HIGH );
return;
}

tmrpcm.play( "a.wav" );    // the sound file "music"
}

void loop( ) {
}

```

ខ. ការចាក់ភ្លេងដោយមិនប្រើ SD card

ការចាក់ភ្លេងដោយ Arduino ផ្ទាល់ យើងមិនអាចចាក់ជាចម្រៀងមនុស្សច្រៀងបានទេ គឺចាក់បានតែ ភ្លេងសុទ្ធច្រប៉ុណ្ណោះ ដោយចេញជាភ្លេងតាម melody note ផ្សេងៗ។ ក្នុងការលេងភ្លេងពិតៗ note ភ្លេង នីមួយៗមានប្រេកង់សម្លេងផ្សេងគ្នា។ យើងឱ្យ Arduino ផលិតប្រេកង់ចេញមកជា note ភ្លេងមួយៗ។ ដើម្បី បានជាភ្លេងចេញមកយើងត្រូវប្រើពាក្យបញ្ជាថ្មីមួយគឺ tone() ដែលជាពាក្យបញ្ជាកំណត់ឱ្យ Arduino ផលិត Square wave ដែលមាន duty cycle 50% តាមប្រេកង់យើងកំណត់។

ទម្រង់ tone()

tone(pin, frequency)

យើងអាចបន្ថែម note ភ្លេងនីមួយៗ ឱ្យវែងបានតាមទម្រង់ខាងក្រោម

tone(pin, frequency, duration)

ដោយ pin គឺជើងរបស់ Arduino ដែលត្រូវកំណត់ឱ្យបង្កើតប្រេកង់ចេញមក frequency គឺប្រេកង់របស់ note ភ្លេងនីមួយៗ និង duration គឺប្រវែងរបស់ note ភ្លេង គិតជាមីលីវិនាទី។ pin ឬជើង Arduino យើង មានច្រើនដែលជាជើង Digital យកជើងមួយណាប្រើក៏បានដែរ។ តើយើងដឹង frequency ឬប្រេកង់របស់ note ភ្លេងផ្សេងៗបានដោយរបៀបណា ? ខាងក្រោមជាតារាងប្រេកង់ note ភ្លេង៖

		Notes											
		C	C#	D	D#	E	F	F#	G	G#	A	A#	B
Octave	0	16	17	18	19	21	22	23	25	26	28	29	31
	1	33	35	37	39	41	44	46	49	52	55	58	62
	2	65	69	73	78	82	87	93	68	104	110	117	124
	3	131	139	147	156	165	175	185	196	208	220	233	247
	4	262	277	294	311	330	349	370	392	415	440	466	494
	5	523	554	587	622	622	699	740	784	831	880	932	988
	6	1047	1109	1175	1245	1319	1397	1480	1568	1661	1760	1865	1976
	7	2093	2217	2349	2489	2637	2794	2960	3136	3322	3520	3729	3951
	8	4186	4435	4699	4978	5274	5588	5920	6272	6645	7040	7459	7902

ឧទាហរណ៍ បង្កើតភ្លេងចេញពី Arduino មាន ៣ភ្លេងដោយប្រើកុងតាក់ជាអ្នកកំណត់ ប្តូរភ្លេងទី១ ឬទី២។

```
#include "pitches.h"

#define melodyPin 9

// note ភ្លេង Jingle Bells
int jingle_melody[] = {
  NOTE_E5, NOTE_E5, NOTE_E5,
  NOTE_E5, NOTE_E5, NOTE_E5,
  NOTE_E5, NOTE_G5, NOTE_C5, NOTE_D5,
  NOTE_E5,
  NOTE_F5, NOTE_F5, NOTE_F5, NOTE_F5,
  NOTE_F5, NOTE_E5, NOTE_E5, NOTE_E5, NOTE_E5,
  NOTE_E5, NOTE_D5, NOTE_D5, NOTE_E5,
  NOTE_D5, NOTE_G5
};

// ចង្វាក់ភ្លេង Jingle bells
int jingle_tempo[] = {
  8, 8, 4,
  8, 8, 4,
  8, 8, 8, 8,
  2,
  8, 8, 8, 8,
  8, 8, 8, 16, 16,
  8, 8, 8, 8,
  4, 4
};

// note ភ្លេង We wish you a merry Christmas
int wish_melody[] = {
  NOTE_B3,
  NOTE_F4, NOTE_F4, NOTE_G4, NOTE_F4, NOTE_E4,
```

```

NOTE_D4, NOTE_D4, NOTE_D4,
NOTE_G4, NOTE_G4, NOTE_A4, NOTE_G4, NOTE_F4,
NOTE_E4, NOTE_E4, NOTE_E4,
NOTE_A4, NOTE_A4, NOTE_B4, NOTE_A4, NOTE_G4,
NOTE_F4, NOTE_D4, NOTE_B3, NOTE_B3,
NOTE_D4, NOTE_G4, NOTE_E4,
NOTE_F4
};

```

```
// ចង្វាក់ភ្លេង We wish you a merry Christmas
```

```
int wish_tempo[] = {
```

```

4,
4, 8, 8, 8, 8,
4, 4, 4,
4, 8, 8, 8, 8,
4, 4, 4,
4, 8, 8, 8, 8,
4, 4, 8, 8,
4, 4, 4,
2

```

```
};
```

```
// note ភ្លេង Santa Claus is coming to town
```

```
int santa_melody[] = {
```

```

NOTE_G4,
NOTE_E4, NOTE_F4, NOTE_G4, NOTE_G4, NOTE_G4,
NOTE_A4, NOTE_B4, NOTE_C5, NOTE_C5, NOTE_C5,
NOTE_E4, NOTE_F4, NOTE_G4, NOTE_G4, NOTE_G4,
NOTE_A4, NOTE_G4, NOTE_F4, NOTE_F4,
NOTE_E4, NOTE_G4, NOTE_C4, NOTE_E4,
NOTE_D4, NOTE_F4, NOTE_B3,
NOTE_C4

```

```
};
```

// ចង្វាក់ភ្លេង Santa Claus is coming to town

```
int santa_tempo[] = {
    8,
    8, 8, 4, 4, 4,
    8, 8, 4, 4, 4,
    8, 8, 4, 4, 4,
    8, 8, 4, 2,
    4, 4, 4, 4,
    4, 2, 4,
    1
};
```

```
int switch1 = 0;
```

```
int switch2 = 0;
```

```
int switch3 = 0;
```

// អថេរភ្លេង សម្រាប់ជ្រើសរើសភ្លេង

```
int song = 0;
```

```
void setup( void ) {
```

```
    pinMode( 9, OUTPUT );      // ដើងទី 9 តជាមួយ Buzzer
```

```
    pinMode( 13, OUTPUT );     // LED on board នឹងភ្លឺនៅពេលដែល ឮសម្លេង
```

```
    pinMode( 2, INPUT );       // ដើងទី 2 តជាមួយកុងតាក់ទី1
```

```
    pinMode( 3, INPUT );       // ដើងទី 3 តជាមួយកុងតាក់ទី2
```

```
    pinMode( 4, INPUT );       // ដើងទី 4 តជាមួយកុងតាក់ទី3
```

```
}
```

```
void loop( ) {
```

```
    switch1 = digitalRead( 2 );
```

```
    switch2 = digitalRead( 3 );
```

```
    switch3 = digitalRead( 4 );
```

```
    if ( switch1 == HIGH ) { sing( 1 ); }      // ប្រសិនបើចុចកុងតាក់ទី 1 នឹងឮភ្លេងពី sing( 1 )
```

```
    else if ( switch2 == HIGH ) { sing( 2 ); } // ប្រសិនបើចុចកុងតាក់ទី 2 នឹងឮភ្លេងពី sing( 2 )
```

```

else if ( switch3 == HIGH ) { sing( 3 ); } // ប្រសិនបើចុចកុងតាក់ទី 3 នឹងឮភ្លេងពី sing( 3 )
} // end loop

```

```

void sing( int s ) {

```

```

    song = s;

```

```

// ប្រសិនបើ song = 1 លេងភ្លេង We wish you a Merry Christmas

```

```

    if ( song == 1 )

```

```

    {

```

```

        Serial.println( " We wish you a Merry Christmas" );

```

```

        int size = sizeof( wish_melody ) / sizeof( int );

```

```

        for ( int thisNote = 0; thisNote < size; thisNote++ ) {

```

```

            // quarter note = 1000 / 4, eighth note = 1000/8, etc.

```

```

            int noteDuration = 1000 / wish_tempo[thisNote];

```

```

            buzz( melodyPin, wish_melody[thisNote], noteDuration );

```

```

            int pauseBetweenNotes = noteDuration * 1.30;

```

```

            delay( pauseBetweenNotes );

```

```

            buzz( melodyPin, 0, noteDuration );

```

```

        }

```

```

    }

```

```

else if ( song == 2 )

```

```

{

```

```

    Serial.println( " Santa Claus is coming to town" );

```

```

    int size = sizeof( santa_melody ) / sizeof( int );

```

```

    for ( int thisNote = 0; thisNote < size; thisNote++ ) {

```

```

        int noteDuration = 900 / santa_tempo[thisNote];

```

```

        buzz( melodyPin, santa_melody[thisNote], noteDuration );

        int pauseBetweenNotes = noteDuration * 1.30;
        delay( pauseBetweenNotes );

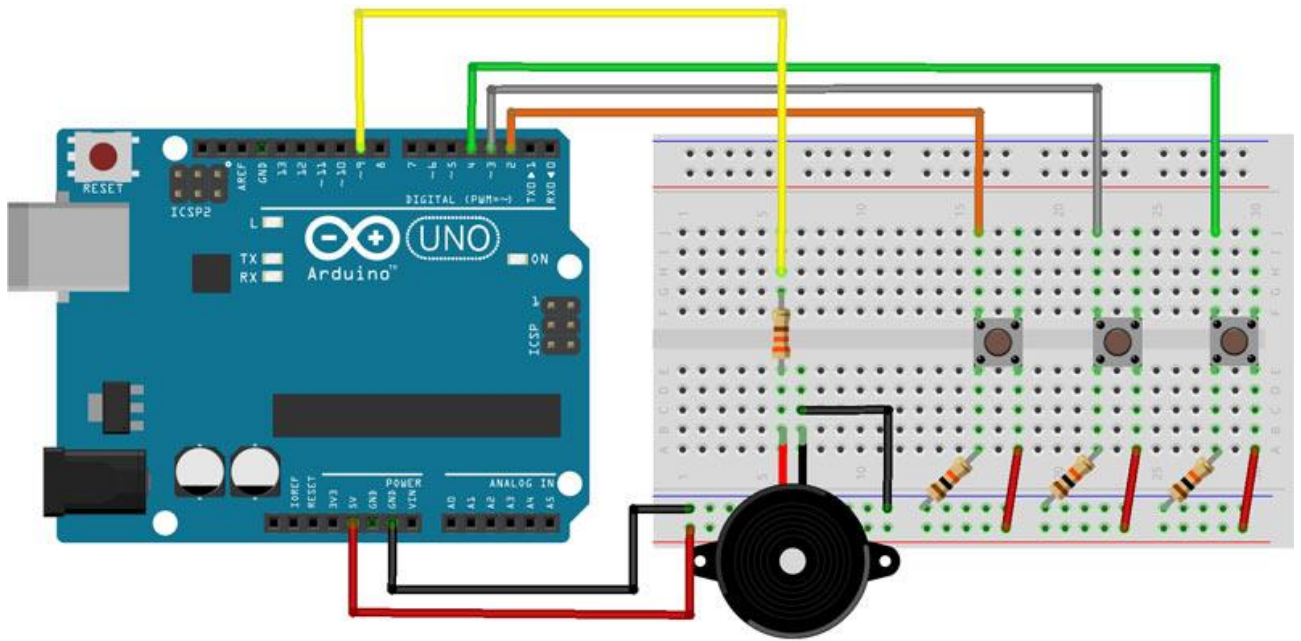
        buzz( melodyPin, 0, noteDuration );
    }
}

else {
    Serial.println( " 'Jingle Bells'" );
    int size = sizeof( jingle_melody ) / sizeof( int );
    for ( int thisNote = 0; thisNote < size; thisNote++ ) {
        int noteDuration = 1000 / jingle_tempo[thisNote];
        buzz( melodyPin, jingle_melody[thisNote], noteDuration );
        int pauseBetweenNotes = noteDuration * 1.30;
        delay( pauseBetweenNotes );
        buzz( melodyPin, 0, noteDuration );
    }
}
// end sing( s )

void buzz( int targetPin, long frequency, long length )
{
    digitalWrite( 13, HIGH );
    long delayValue = 1000000 / frequency / 2;
    long numCycles = frequency * length / 1000;
    for ( long i = 0; i < numCycles; i++ )
    {
        digitalWrite( targetPin, HIGH );
        delayMicroseconds( delayValue );
        digitalWrite( targetPin, LOW );
        delayMicroseconds( delayValue );
    }
}

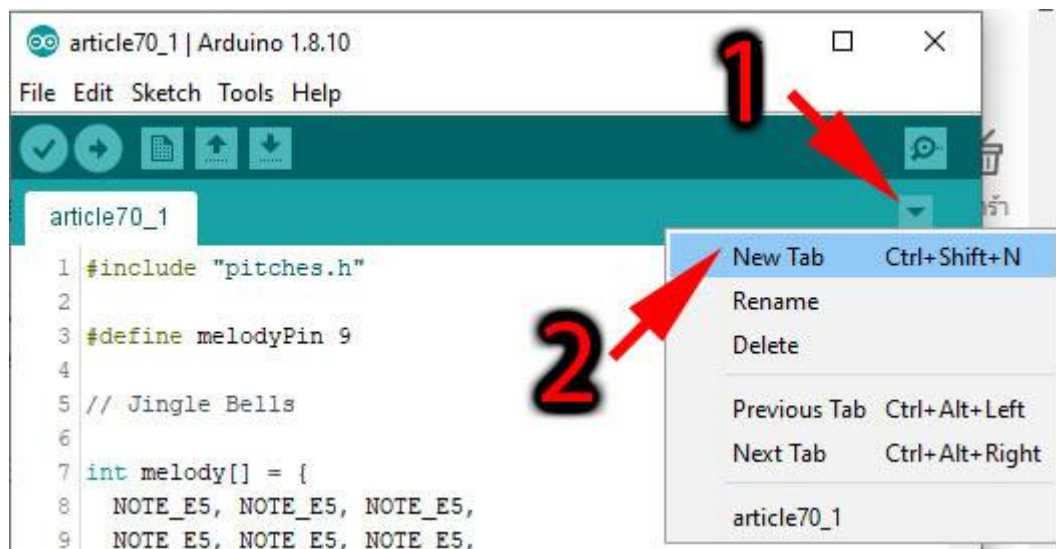
```

```
digitalWrite( 13, LOW);  
}  
// end buzz
```



រូបទី៧.៦ ការតម្លៃគ្លីសម្រាប់ប្រើក្នុងឧទាហរណ៍

ក្នុងឧទាហរណ៍នេះមានបន្ថែម tab ថ្មីមួយឈ្មោះថា pitches.h ហើយយើងធ្វើការបន្ថែមដោយចុច ត្រង់សញ្ញាព្រួញខាងស្តាំ រួចយកពាក្យថា New Tab ឬចុច Ctrl+Shift+N ។



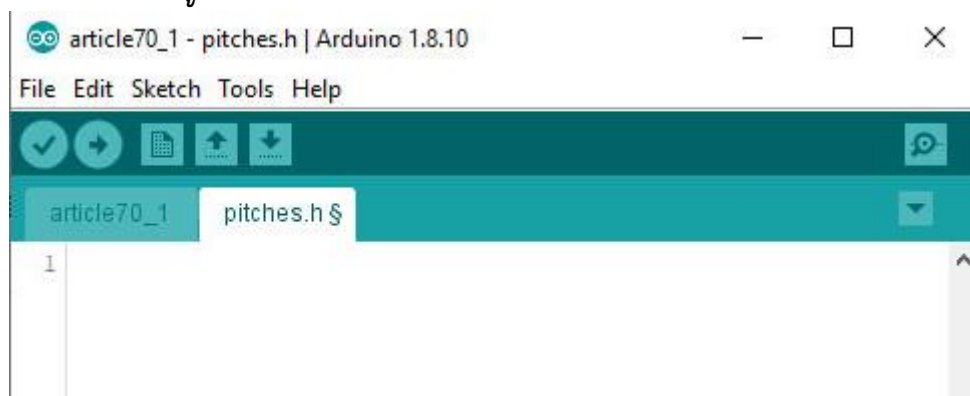
រូបទី៧.៧ ការជ្រើសយក New Tap

នៅជួរខាងក្រោមមានពណ៌លឿង និងមានកន្លែងឱ្យដាក់ឈ្មោះ យើងត្រូវដាក់ " pitches.h " ចូល ហើយចុច OK (យើងត្រូវដាក់ទាំង .h ចូលដែរ ប្រសិនបើមិន .h ចូលទេ វានឹងត្រូវប្តូរទៅជា .ino ធ្វើឱ្យមិន អាច include "pitches.h" ដោយសារវាគ្មាន file)



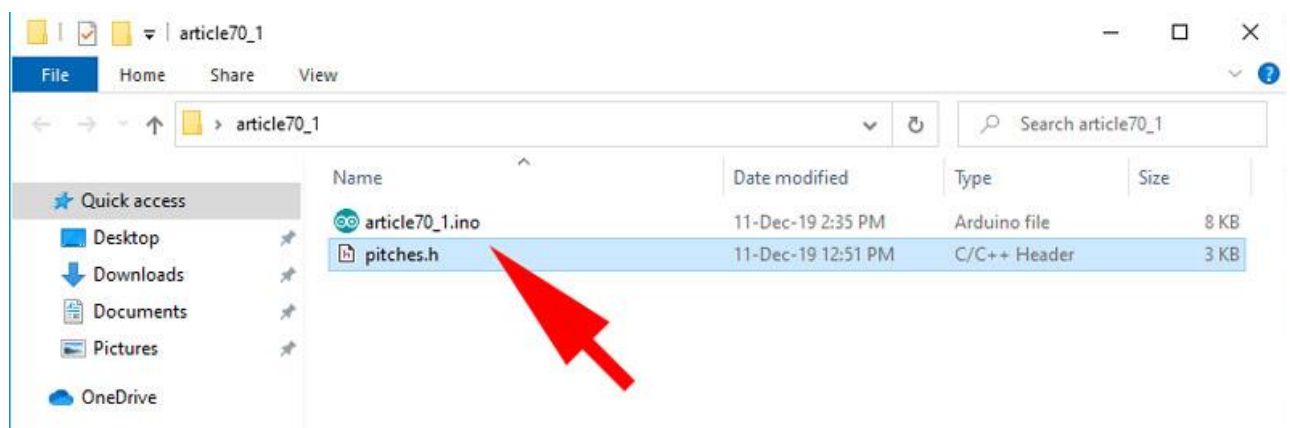
រូបទី៧.៨ ការបញ្ចូល pitches.h

ពេលបញ្ចូល pitches.h រួច យើងនឹងបាន Tab ថ្មីមួយទៀតចេញមក



រូបទី៧.៩ Tab pitches.h ថ្មី

ប្រសិនបើត្រូវការប្រើ pitches.h នៅលើគម្រោងផ្សេងទៀត យើងអាច Copy file នេះទៅដាក់កន្លែង ផ្សេងនោះបាន។



រូបទី៧.១០ file pitches.h

មេរៀនទី៨ Basic Wireless Communication

១. IR Remote Control

IR Remote Control គឺជាឧបករណ៍បញ្ជា (តេលេ) ដែលយើងប្រើរាល់ថ្ងៃជាមួយម៉ាស៊ីនត្រជាក់ ទូរទស្សន៍ឬគ្រឿងបញ្ជាផ្សេងៗ។ ក្នុងមេរៀននេះយើងនឹងសិក្សាអំពីការចម្លង ក៏ដូចជាការបង្កើតតេលេនេះ ជាមួយ Arduino ។ IR Remote Control ក៏មិនខុសពី sensor ផ្សេងទៀតដែរ ដើម្បីអាចប្រើបានយើងត្រូវ ការ Library មួយដែលរកបានតាមរយៈ៖

<https://github.com/z3t0/Arduino-IRremote>



រូបទី៨.១ sensor IR Remote Control

ក. Receiver

- Constructor

ទម្រង់

```
IRrecv( IR pin );
```

Example:

```
IRrecv irrecv( 10 );
```

- Decode variable

ទម្រង់

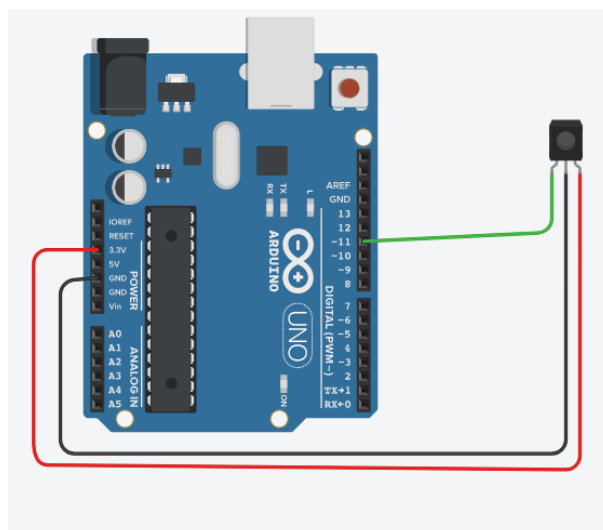
```
decode_results results;
```

ឧទាហរណ៍

```
IRrecv irrecv( 10 );
```

```
decode_results results;
```

- **Begin**
ទម្រង់
`Irrecv.enableIRIn( );`
- **decode**
ទម្រង់
`Irrecv.decode( &results );`
Return:
True : if IR receive data from remote
False : if No data receive
- **Get Value**
ទម្រង់
`results.value;`
Return:
Data from remote
- **Resume**
ទម្រង់
`Irrecv.resume( );`
ឧទាហរណ៍
`Irrecv.resume( );`



រូបទី៨.២ ការតភ្ជាប់ IR Remote ជាមួយ Arduino

ឧទាហរណ៍

```
#include <IRremote.h>
#include<IRremoteInt.h>
    IRrecv irrecv( 10 );
    decode_results results;
void setup( ) {
    Serial.begin( 9600 );
    irrecv.enableIRIn( );
}
void loop( ){
```

```

    if ( irrecv.decode( &results ) ) {
        Serial.println( results.value,HEX );
        irrecv.resume( );
    }
}

```

២. Transmitter

- Constructor

Syntax:

IRsend irsend;

- Send

Syntax:

Irsend.sendSony(data, bits);

Parameter:

Data : data to send

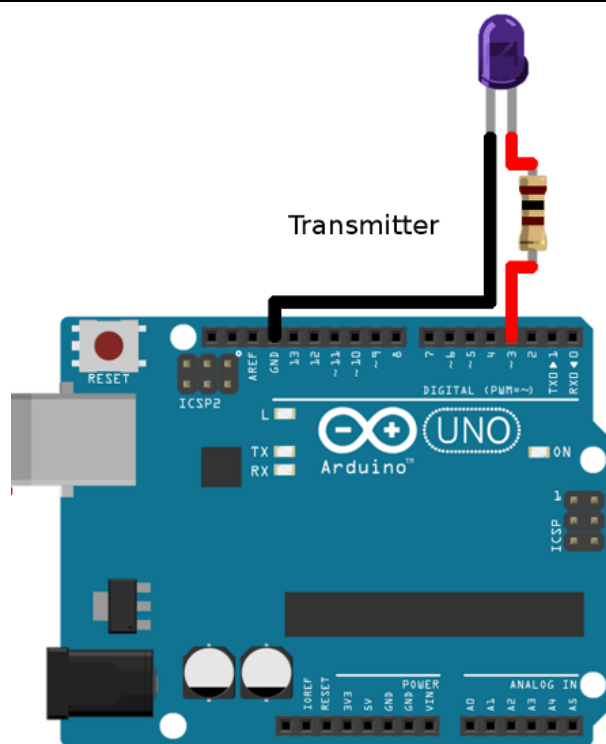
Bits : Number of bit to send

ខាងក្រោមនេះជាសៀគ្វី និងគំរូកូដសម្រាប់ធ្វើការបញ្ជូនពី Arduino ទៅកាន់ឧបករណ៍ណាមួយរបស់យើងដែលប្រើ៖

Hardware specifications

Board /CPU	Send Pin	Timers
ATtiny84	6	1
ATtiny55	1	TINYO
ATmega8	9	1
Atmega32u4	5,9,13	1,3,4
ATmegaa48, ATmega88, ATmega168, ATmega328	3,9	1,2
ATmega1284	13,14,6	1,2,3
ATmega164, ATmega324, ATmega644	13,14	1,2
ATmega8535 ATmega16, ATmega32	13	1
ATmega64, ATmega128	13	1
ATmega1280, ATmega2560	5,6,9,11,46	1,2,3,4,5
ESP32	N/A(not supported)	1

Teensy 1.0	17	1
Teensy2.0	9, 10, 14	1, 3, 4_HS
Teensy++ 1.0/2.0	1, 16, 25	1, 2, 3
Teensy 3.0/3.1	5	CMT
Teensy-LC	16	TPM1



រូបទី៨.៣ ការតភ្ជាប់ IR Transceiver ជាមួយ Arduino

```
#include<IRremote.h>
IRsend irsend;
void setup( )
{
}
void loop( ){
  for ( int i= 0; i < 3;i++){
    irsend.sendSony( 0xa90,12);
    delay( 5000 );
  }
  delay( 5000 );
}
```

ឧទាហរណ៍១ ចូរសរសេរកូដបញ្ជា LED ១គ្រាប់ តាមរយៈតេលេឡិភ្លឺ និងរលត់បាន

```
#include <IRremote.h>
#include <IRremoteInt.h>
IRrecv irrecv( 45 );
decode_results results;
void setup( ) {
    irrecv.enableIRIn( );
    pinMode( 2,OUTPUT );
}
void loop( ) {
    if( irrecv.decode( &results ) ){
        if ( results.value==0xFF00001A ) {
            digitalWrite( 2,1 );
        } else {
            digitalWrite( 2,0 );
        }
        irrecv.resume( );
    }
}
```

ដោយសារ remote មានច្រើនប្រភេទលើទីផ្សារ និងមានកូដរបស់ Key នីមួយៗផ្សេងគ្នា ដូចនេះត្រូវមានកូដរួមមួយដើម្បីឆែកមើលកូដរបស់ Key នោះៗ។ library IRremote គឺជាអ្នកអានស៊ីញ៉ាល់ និងបញ្ចេញជាកូដរបស់ Key មួយៗ ក្នុងទម្រង់ជាលេខគោល ១៦ ដោយអាស្រ័យថាយើងចុច Key មួយណា។ យើងធ្វើការបញ្ចេញកូដរបស់ Key នៅលើ Serial Monitor តាមរយៈកូដខាងក្រោម៖

```
#include <IRremote.h>
#define RECEIVER_PIN 2
IRrecv receiver( RECEIVER_PIN );
decode_results results;
void setup( ) {
    Serial.begin( 9600 );
    receiver.enableIRIn( ); }
void loop( ) {
    if ( receiver.decode( &results ) ) {
        Serial.println( results.value, HEX );
```

```

        receiver.resume( );
    }
}

```

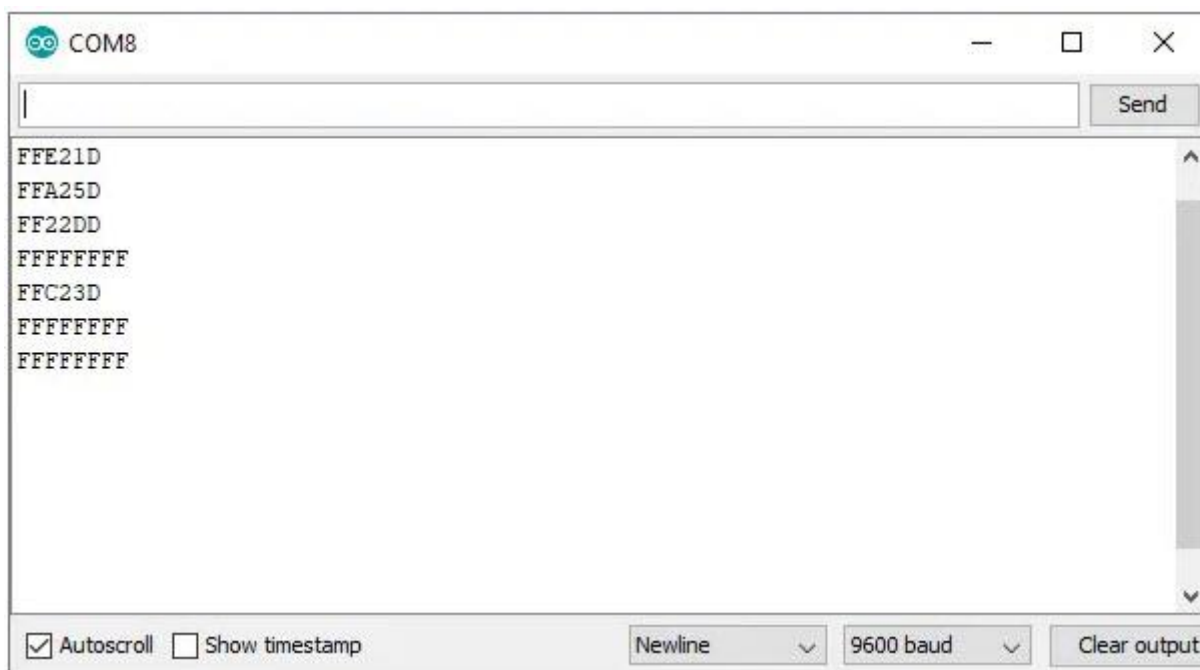
Key	Code
POWER	0xFD00FF
VOL+	0xFD807F
FUNC/STOP	0xFD40BF
◀◀	0xFD20DF
▶▶	0xFDA05F
▶▶	0xFD609F
▼	0xFD10EF
VOL-	0xFD906F
▲	0xFD50AF
0	0xFD30CF
EQ	0xFDB04F
ST/REPT	0xFD708F
1	0xFD08F7
2	0xFD8877
3	0xFD48B7
4	0xFD28D7
5	0xFDA857
6	0xFD6897
7	0xFD18E7
8	0xFD9867
9	0xFD58A7

សម្រាប់កូដរបស់ Key remote ដែលប្រើនៅថ្ងៃនេះ
មានដូចក្នុងតារាងខាងឆ្វេង



រូបទី៨.៤ remote

ក្រោយពីយើង Upload កូដចូលហើយយើងបើក Serial Monitor យើងត្រូវចុច Key នៅលើ remote ពេលនេះយើងនឹងបាន កូដរបស់ Key នីមួយៗ ជាលេខគោល ១៦។



រូបទី៨.៥ ការបង្ហាញទិន្នន័យកូដរបស់ Key remote

ឧទាហរណ៍២ ចូរសរសេរកូដបង្ហាញឈ្មោះរបស់ Key នីមួយៗនៅលើ Serial Monitor។

```
#include <IRremote.h>

#define RECEIVER_PIN 2          // define the IR receiver pin

IRrecv receiver( RECEIVER_PIN ); // create a receiver object of the IRrecv class
decode_results results; // create a results object of the decode_results class
unsigned long key_value = 0;    // variable to store the key value

void setup( ) {
  Serial.begin( 9600 );
  receiver.enableIRIn( );      // enable the receiver
}

void loop( ) {
  if ( receiver.decode( &results ) ) {
    if ( results.value == 0xFFFFFFFF ) { // if the value is equal to 0xFFFFFFFF
      results.value = key_value;        // set the value to the key value
    }
    switch ( results.value ) {          // compare the value to the following cases
```

```

case 0xFD00FF:           // if the value is equal to 0xFD00FF
    Serial.println("POWER"); // print "POWER" in the Serial Monitor
    break;
case 0xFD807F:
    Serial.println("VOL+");
    break;
case 0xFD40BF:
    Serial.println("FUNC/STOP");
    break;
case 0xFD20DF:
    Serial.println("<<");
    break;
case 0xFDA05F:
    Serial.println(">||");
    break ;
case 0xFD609F:
    Serial.println(">>|");
    break ;
case 0xFD10EF:
    Serial.println("DOWN");
    break ;
case 0xFD906F:
    Serial.println("VOL-");
    break ;
case 0xFD50AF:
    Serial.println("UP");
    break ;
case 0xFD30CF:
    Serial.println("0");
    break ;
case 0xFDB04F:
    Serial.println("EQ");
    break ;

```

```

case 0xFD708F:
    Serial.println( "ST/REPT" );
    break ;
case 0xFD08F7:
    Serial.println( "1" );
    break ;
case 0xFD8877:
    Serial.println( "2" );
    break ;
case 0xFD48B7:
    Serial.println( "3" );
    break ;
case 0xFD28D7:
    Serial.println( "4" );
    break ;
case 0xFDA857:
    Serial.println( "5" );
    break ;
case 0xFD6897:
    Serial.println( "6" );
    break ;
case 0xFD18E7:
    Serial.println( "7" );
    break ;
case 0xFD9867:
    Serial.println( "8" );
    break ;
case 0xFD58A7:
    Serial.println( "9" );
    break ; }
key_value = results.value;      // store the value as key_value
receiver.resume( );            // reset the receiver for the next code
} }

```

ឧទាហរណ៍៣ ចូរសរសេរកូដបង្ហាញឈ្មោះរបស់ Key នីមួយៗនៅលើ LCD ។

```

#include <IRremote.h>    // include the IRremote library
#include <LiquidCrystal.h> // include the LiquidCrystal library
LiquidCrystal lcd = LiquidCrystal( 2, 3, 4, 5, 6, 7 );
//create an LCD object with pin parameters ( RS, E, D4, D5, D6, D7 )

#define RECEIVER_PIN 8 // define the IR receiver pin
IRrecv receiver( RECEIVER_PIN ); // create a receiver object of the IRrecv class
decode_results results;          // create a results object of the decode_results class
unsigned long key_value = 0;     // variable to store the key value

void setup( ) {
  lcd.begin( 16, 2 );           // enable LCD with 16 columns and 2 rows
  receiver.enableIRIn( );      // enable the receiver
}

void loop( ) {
  if ( receiver.decode( &results ) ) { // decode the received signal and store it in results
    if ( results.value == 0xFFFFFFFF ) { // if the value is equal to 0xFFFFFFFF
      results.value = key_value; // set the value to the key value
    }
    switch ( results.value ) {      // compare the value to the following cases
      case 0xFD00FF:                // if the value is equal to 0xFD00FF
        lcd.clear( );              // clear the display
        break;
      case 0xFD807F:
        lcd.print( "VOL+" );
        break;
      case 0xFD40BF:
        lcd.print( "FUNC/STOP" );
        break;
      case 0xFD20DF:
        lcd.print( "|<<" );

```

```

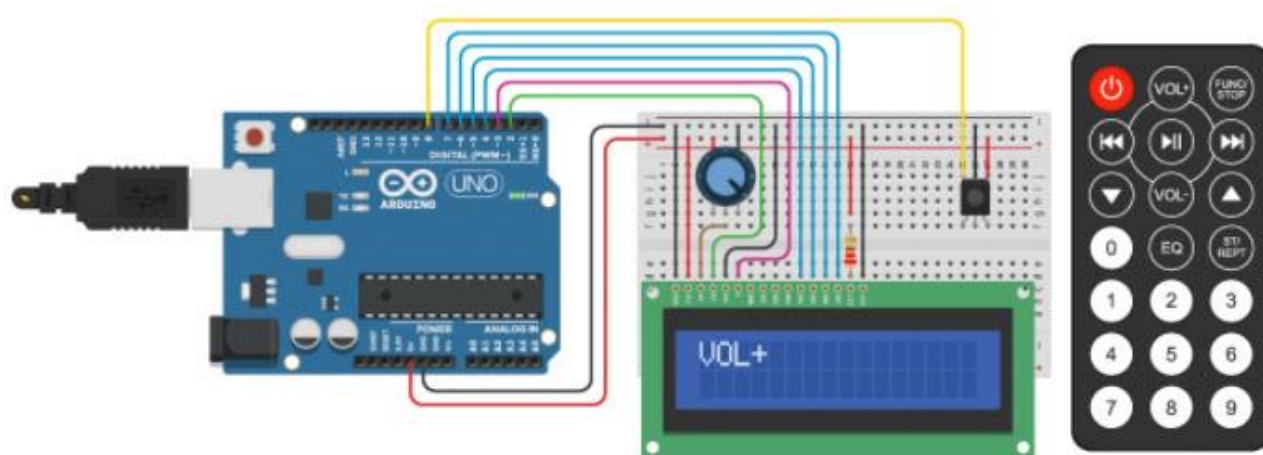
        break;
case 0xFDA05F:
    lcd.print(">||");
    break ;
case 0xFD609F:
    lcd.print(">>|");
    break ;
case 0xFD10EF:
    lcd.print("DOWN");
    break ;
case 0xFD906F:
    lcd.print("VOL-");
    break ;
case 0xFD50AF:
    lcd.print("UP");
    break ;
case 0xFD30CF:
    lcd.print("0");
    break ;
case 0xFDB04F:
    lcd.print("EQ");
    break ;
case 0xFD708F:
    lcd.print("ST/REPT");
    break ;
case 0xFD08F7:
    lcd.print("1");
    break ;
case 0xFD8877:
    lcd.print("2");
    break ;
case 0xFD48B7:
    lcd.print("3");

```

```

        break ;
    case 0xFD28D7:
        lcd.print("4");
        break ;
    case 0xFDA857:
        lcd.print("5");
        break ;
    case 0xFD6897:
        lcd.print("6");
        break ;
    case 0xFD18E7:
        lcd.print("7");
        break ;
    case 0xFD9867:
        lcd.print("8");
        break ;
    case 0xFD58A7:
        lcd.print("9");
        break ;
    }
    key_value = results.value;      // store the value as key_value
    receiver.resume( );            // reset the receiver for the next code
}
}

```



រូបទី៨.៦ ការតភ្ជាប់គ្នារវាង LCD និង IRremote តាមរយៈ Arduino Uno (ប្រើក្នុងឧទាហរណ៍ទី២ និងទី៣)

២. UART Tx/Rx Transceiver module (HC-12)

Arduino ក៏អនុញ្ញាតឱ្យយើងអាចបង្កើត Serial Communication តាមរយៈជើងផ្សេងៗដទៃទៀត ដែលមិនមែនជើង Tx/Rx របស់វាបានដោយប្រើ Software Serial។



រូបទី៨.៧ HC-12

HC-12 មានភ្ជាប់មកជាមួយនូវ AT ដែលអាចប្រើសម្រាប់ការ Register ។ ដើម្បីងាយស្រួលប្រើការ យើងត្រូវមាន Library ដែលអាចទាញយកពីក្នុងគេហទំព័រ <https://github.com/daar/HC-12> ។ នៅក្នុង HC-12 ក៏មានទម្រង់មួយចំនួនដែលត្រូវប្រើផងដែរដូចជា៖

- AT សម្រាប់តេស្តមើល Communication រវាង Arduino និង HC-12 module

Return:

OK មានន័យថាការតភ្ជាប់បានជោគជ័យ ផ្ទុយទៅវិញគឺគ្មានអ្វី return

- AT+Bxxxx សម្រាប់កំណត់ Baudrate

ទម្រង់

AT+Bxxxx

Return:

Ok+C002

ឧទាហរណ៍

AT+C002

- AT+Px សម្រាប់កំណត់អនុភាពបញ្ជូនចេញឱ្យវា ដែលមានតម្លៃពី 1-8 = -1dBm to 20dBm

ទម្រង់

AT+Px

Return:

OK+P5

ឧទាហរណ៍

AT+P5

- **AT+Uxxx** សម្រាប់កំណត់ចំនួន bit, Parity and Stop bit ដែល o មានន័យថា 0pp, E មានន័យថា Even, 1 stop bit 1, 2 stop bit 2 and 3 stop bit 1.5

ទម្រង់

AT+Uxxx

Return:

OK+U8E1

ឧទាហរណ៍

AT+ U8E1

- **AT+FUx** សម្រាប់កំណត់ mode ឱ្យ Serial។ មាន៤ ម៉ូតគឺ FU1, FU2, FU3 and FU4។
FU1: ប្រើសម្រាប់សន្សំថាមពល ដែល HC-12 ស៊ីត្រីមតែ 3.3mA ប្រើបានគ្រប់ Baud rate in air គឺបើស្មើ 250000 bps ។
FU2: សម្រាប់សន្សំថាមពលបានកាន់តែច្រើនជាមុន គឺ 80uA មានតែ Baud rate 1200bps, 2400bps, 4800bps ទេដែលប្រើបាន។ ដើម្បីជៀសវាង ការបាត់បង់ទិន្នន័យតាមផ្លូវ ដោយទាមទារឱ្យការបញ្ជូនត្រូវចំណាយពេលយូរជាង ១វិនាទី។
FU3: ស៊ីថាមពល 16mA និងប្រើបានគ្រប់ Baud rate តែ Baud rate in air ស្មើ Baud rate ដែលយើងដាក់បញ្ចូលតាម AT+Bxxx។
FU4: សម្រាប់បញ្ជូនរយៈចម្ងាយឆ្ងាយរហូតដល់ 1.8Km។ ដើម្បីជៀសវាង ការបាត់បង់ទិន្នន័យតាមផ្លូវ ដែលទាមទារឱ្យការបញ្ជូនត្រូវចំណាយពេលយូរជាង ២វិនាទី។

ទម្រង់

AT+FUx

Return:

AT+OK

ឧទាហរណ៍

AT+FU4

- **AT+DEFAULT** សម្រាប់កំណត់ឱ្យវាទៅកាន់ Default value ដើមពីរោងចក្រវិញ។

ទម្រង់

AT+DEFAULT

Return:

OK+DEFAULT

ឧទាហរណ៍

AT+DEFAULT

ឧទាហរណ៍ទី១ ចូរសរសេរកូដបញ្ជាអំពូល LED ១គ្រាប់ ឱ្យភ្លឺ ឬរលត់ ដោយបញ្ជាតាមរយៈ Volume Resistor។

HC-12 Transmitter

/*

Module // Arduino UNO/NANO

GND -> GND

Vcc -> 3.3V

Tx -> D10

Rx -> D11

*/

#include <SoftwareSerial.h>

SoftwareSerial HC12(10, 11); // HC-12 TX Pin, HC-12 RX Pin

int pot = A2;

void setup() {

Serial.begin(9600); // Serial port to computer

HC12.begin(9600); // Serial port to HC12

pinMode(pot,INPUT);

}

void loop() {

int val = map(analogRead(pot),0,1024,0,255);

HC12.write(val); // Send that data to HC-12

}

HC-12 Receiver

/*

Module // Arduino UNO/NANO

GND -> GND

Vcc -> 3.3V

Tx -> D10

Rx -> D11

*/

```

#include <SoftwareSerial.h>

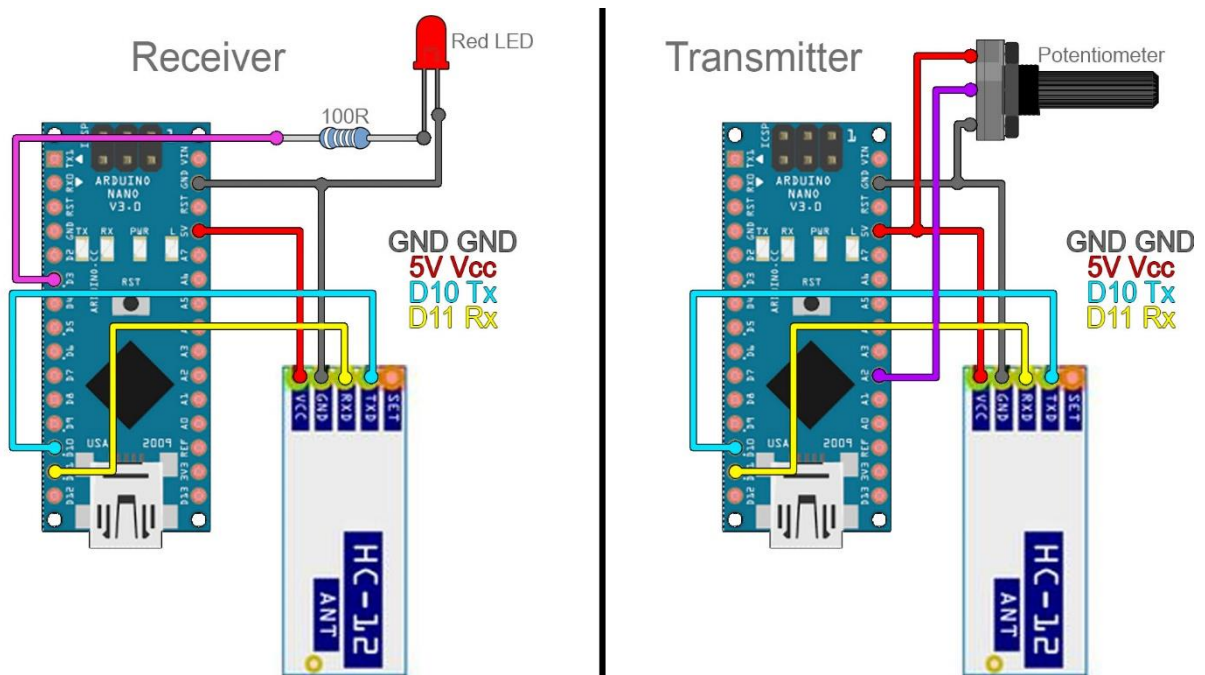
SoftwareSerial HC12( 10, 11 ); // HC-12 TX Pin, HC-12 RX Pin

int LED = 3;

void setup( ) {
    Serial.begin( 9600 );          // Serial port to computer
    HC12.begin( 9600 );           // Serial port to HC12
    pinMode( LED, OUTPUT );
}

void loop( ) {
    while ( HC12.available( ) ) { // If HC-12 has data
        int val = HC12.read( );
        Serial.println( HC12.read( ) ); // Send the data to Serial monitor
        analogWrite( LED, val );
    }
}

```



រូបទី៨.៨ សៀគ្វីការត HC-12 ជាមួយ board Nano ទាំង Receiver និង Transmitter (ប្រើក្នុងឧទាហរណ៍ទី១)

ឧទាហរណ៍ទី២ ចូរសរសេរកូដបញ្ជា Servo Motor ដោយប្រើ Joystick Module ជាអ្នកបញ្ជា ហើយប្រើ RF24 ជាអ្នកភ្ជាប់។

HC-12 Transmitter

```

#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#define led 12
RF24 radio( 7, 8 ); // CE, CSN
const byte addresses[][6] = {"00001", "00002"};
boolean buttonState = 0;
void setup( ) {
    pinMode( 12, OUTPUT );
    radio.begin( );
    radio.openWritingPipe( addresses[1] ); // 00002
    radio.openReadingPipe( 1, addresses[0] ); // 00001
    radio.setPALevel( RF24_PA_MIN );
}
void loop( ) {
    delay( 5 );
    radio.stopListening( );
    int potValue = analogRead( A0 );
    int angleValue = map( potValue, 0, 1023, 0, 180 );
    radio.write( &angleValue, sizeof( angleValue ) );
    delay( 5 );
    radio.startListening( );
    while ( !radio.available( ) );
    radio.read( &buttonState, sizeof( buttonState ) );
    if ( buttonState == HIGH ) {
        digitalWrite( led, HIGH );
    }
    else {
        digitalWrite( led, LOW );
    }
}

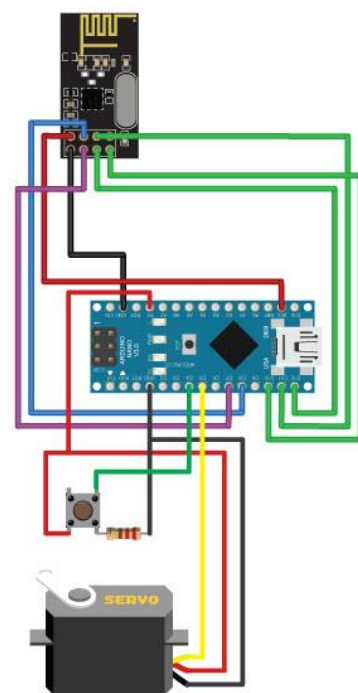
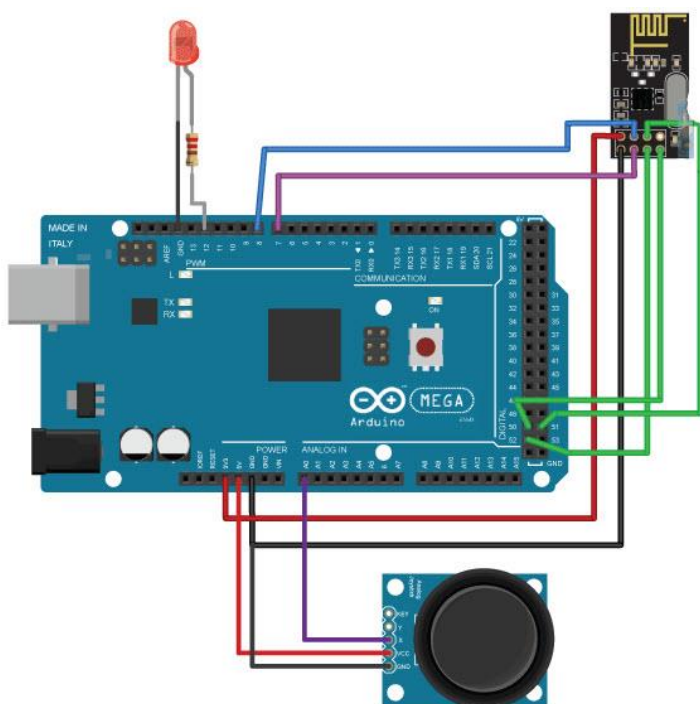
```

HC-12 Receiver

```

#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
RF24 radio( 7, 8 ); // CE, CSN
const byte address[6] = "00001";
void setup( ) {
    Serial.begin( 9600 );
    radio.begin( );
    radio.openReadingPipe( 0, address );
    radio.setPALevel( RF24_PA_MIN );
    radio.startListening( );
}
void loop( ) {
    if ( radio.available( ) ) {
        char text[32] = "";
        radio.read( &text, sizeof( text ) );
        Serial.println( text );
    }
}

```



រូបទី៨.៩ សៀគ្វីការភ្ជាប់ RT24 ជាមួយ board Nano ទាំង Receiver និង Transmitter (ប្រើក្នុងឧទាហរណ៍ទី២)

ឧទាហរណ៍ទី៣ ចូរធ្វើកូនឡានបញ្ជាដោយ Joystick Module ។**HC-12 Transmitter**

```

int xAxis, yAxis;
int GuideLaserbutton = A2;
int GuideLaser;
void setup( ) {
  Serial.begin( 9600 );
  pinMode( GuideLaserbutton, INPUT );
  digitalWrite( GuideLaserbutton, HIGH );
}
void loop( ) {
  xAxis = analogRead( A0 );      // Read Joysticks X-axis
  yAxis = analogRead( A1 );      // Read Joysticks Y-axis
  GuideLaser = digitalRead( GuideLaserbutton );      //Turning on/off the led laser
  Serial.write( xAxis / 4 ); // Dividing by4 for converting from 0-1023 to 0 - 256, ( 1 byte ) range
  Serial.write( yAxis / 4 );
  Serial.write( digitalRead( GuideLaser ) );
  delay( 30 );
}

```

HC-12 Receiver

```

int x_pos;
int y_pos;
int x = 0;
int y = 0;
//Motor Pins
int EN_A = 11;      //Enable pin for first motor
int IN1 = 9;        //control pin for first motor
int IN2 = 8;        //control pin for first motor
int IN3 = 7;        //control pin for second motor
int IN4 = 6;        //control pin for second motor
int EN_B = 10;      //Enable pin for second motor

```

```

int motor_speed1;
int motor_speed2;
//GuideLaser
int GuideLaser;
int GuideLaserP = 4;      //control pin for GuideLaser
void setup ( ) {
    Serial.begin ( 9600 );    //Starting the serial communication at 9600 baud rate
                                //Initializing the motor pins as output
    pinMode( EN_A, OUTPUT );
    pinMode( IN1, OUTPUT );
    pinMode( IN2, OUTPUT );
    pinMode( IN3, OUTPUT );
    pinMode( IN4, OUTPUT );
    pinMode( EN_B, OUTPUT );
    pinMode( GuideLaserP, OUTPUT );
}
void loop ( ) {
    x_pos = 510;
    y_pos = 510;
    while ( Serial.available( ) == 0 ) {}
    x = Serial.read( );
    delay( 10 );
    y = Serial.read( );
    delay( 10 );
    GuideLaser = Serial.read( );
    delay( 10 );
    if ( GuideLaser == HIGH ) {
        digitalWrite( GuideLaserP, HIGH );    //LED control part.
    }
    else {
        digitalWrite( GuideLaserP, LOW );
    }
    x_pos = x * 4;

```

```

y_pos = y * 4;
if ( x_pos < 400 ) {                                //Rotating the left motor in clockwise direction
    motor_speed1 = map( x_pos, 400, 0, 0, 255 );
                                           //Mapping the values to 0-255 to move the motor
    digitalWrite( IN1, LOW );
    digitalWrite( IN2, HIGH );
    analogWrite( EN_A, motor_speed1 );
}

else if ( x_pos > 400 && x_pos < 600 ) {
                                           //Motors will not move when the joystick will be at center
    digitalWrite( IN1, LOW );
    digitalWrite( IN2, LOW );
}

else if ( x_pos > 600 ) {                        //Rotating the left motor in anticlockwise direction
    motor_speed1 = map( x_pos, 600, 1023, 0, 255 );
    digitalWrite( IN1, HIGH );
    digitalWrite( IN2, LOW );
    analogWrite( EN_A, motor_speed1 );
}

if ( y_pos < 400 ) {                            //Rotating the right motor in clockwise direction
    motor_speed2 = map( y_pos, 400, 0, 0, 255 );
    digitalWrite( IN3, LOW );
    digitalWrite( IN4, HIGH );
    analogWrite( EN_B, motor_speed2 );
}

else if ( y_pos > 400 && y_pos < 600 ) {
    digitalWrite( IN3, LOW );
    digitalWrite( IN4, LOW );
}

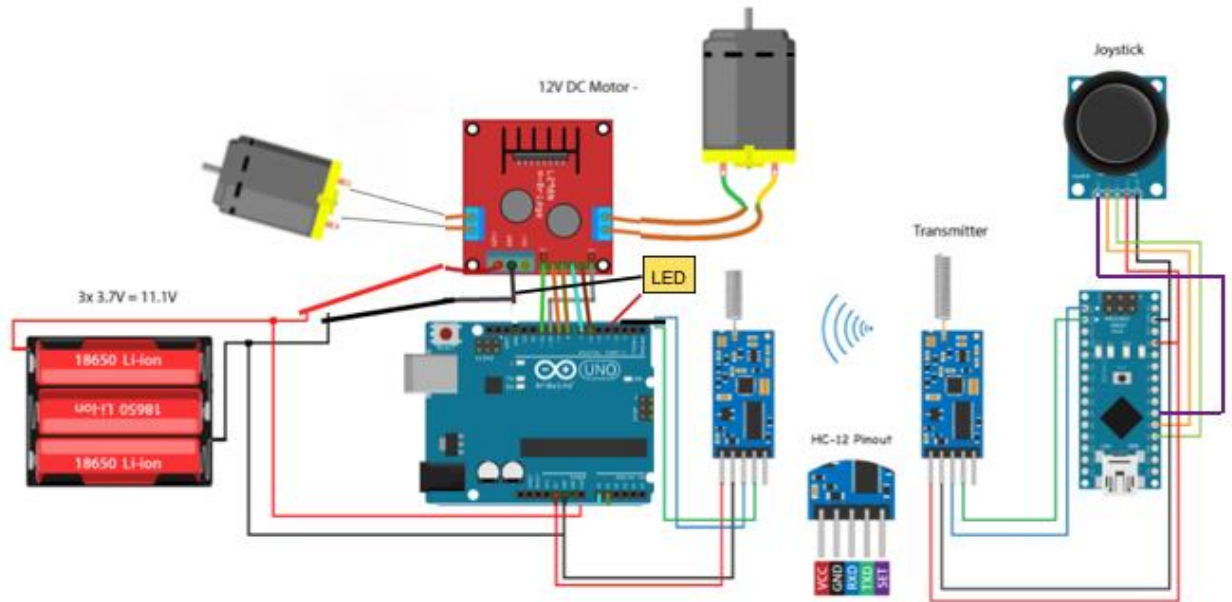
else if ( y_pos > 600 ) {                        //Rotating the right motor in anticlockwise direction
    motor_speed2 = map( y_pos, 600, 1023, 0, 255 );
    digitalWrite( IN3, HIGH );

```

```

digitalWrite( IN4, LOW );
analogWrite( EN_B, motor_speed2 );
}
}

```



រូបទី៨.១០ សៀគ្វីការត ប្រើក្នុងឧទាហរណ៍ទី៣ ទាំង Receiverនិង Transmitter

មេរៀនទី៩ Basic Motor control

១. Analog Input/Output

ក. Analog Read

យើងប្រើ Analog Read សម្រាប់ការអានយក និងបញ្ចេញសញ្ញា Analog ក្នុង Arduino ។ ទោះបី Arduino ជា Microcontroller ក៏ដោយ ក៏វាមិនប្រាកដថាប្រើបានតែជាមួយ Digital ដែរ។ នៅក្នុងមេរៀននេះ យើងនឹងសិក្សាពីការប្រើប្រាស់ Analog Pin។ នៅក្នុង Board Arduino Uno មានជើង Analog ចំនួន ៦ជើង ហើយនៅក្នុង Board Arduino Mega មានជើង Analog ចំនួន ១៦ជើង។

Analog reference ប្រើសម្រាប់កំណត់តម្លៃតង់ស្យុង ដែលត្រូវយកទៅធៀបជាមួយតង់ស្យុងនៃជើង Analog Input។ តម្លៃតង់ស្យុងដែលយើងកំណត់នេះជាតម្លៃតង់ស្យុងអតិបរិមាដែលជើង Analog Input អាចទទួលបាន មានន័យថាតម្លៃនោះគឺជា Digital ស្មើ 1023។

ទម្រង់:

analogReference(type)

Parameter:

Type: ជ្រើសរើសប្រភេទដែលត្រូវប្រើ (DEFAULT, INTERNAL, INTERNAL1V1, INTERNAL2V6, EXTERNAL)

- DEFAULT: តង់ស្យុងប្រៀបធៀបមានតម្លៃ 5volts (on 5v Arduino boards) ឬ 3.3volts (on 3.3v Arduino boards)
- INTERNAL: តង់ស្យុងធៀបស្មើ 1.1volts ចំពោះ ATmega328 និង 2.56volts នៅលើ ATmega8 (សម្រាប់ Arduino Mega ពុំមានជម្រើសនេះទេ)
- INTERNAL1V1 តង់ស្យុងធៀបស្មើ 1.1v សម្រាប់ Arduino mega តែប៉ុណ្ណោះ
- INTERNAL2V56 តង់ស្យុងធៀបស្មើ 2.5 សម្រាប់ Arduino mega តែប៉ុណ្ណោះ
- EXTERNAL តង់ស្យុង ដាក់ទៅកាន់ជើង AREF របស់ Arduino ពីខាងក្រៅ (0-5v only) ត្រូវប្រើជាតង់ស្យុងប្រៀបធៀប

Analog Read ប្រើសម្រាប់ទាញយកតម្លៃ Analog ហើយបំប្លែងជាតម្លៃ Digital ។ Resolution Analog to Digital មាន 10 bits ដែលមានន័យថាយើងបំបែកតង់ស្យុងពី 0v- 5v ឱ្យទៅជាតម្លៃលេខចន្លោះពី 0 – 1023 ស្មើនឹង 4.9mV/unit ។

ទម្រង់:

analogRead(pin)

Parameter:

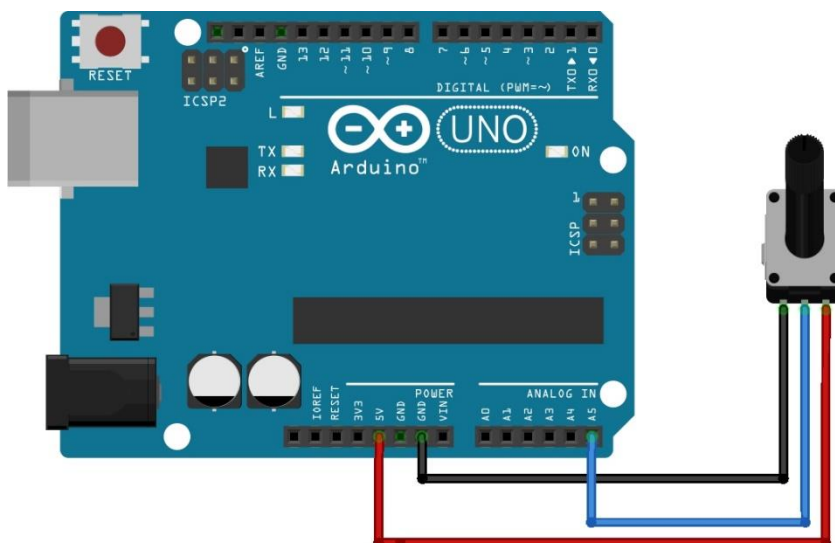
Pin

លេខជើង Analog របស់ Arduino

Return:

0-1023

ឧទាហរណ៍១ ចូរសរសេរកូដប្រើ volume ដើម្បីតេស្តជាមួយ Analog read តាមរយៈការតភ្ជាប់ដូចក្នុងរូបទី៩.១ ហើយបង្ហាញតម្លៃដែលអានបាននោះក្នុង Serial monitor ។



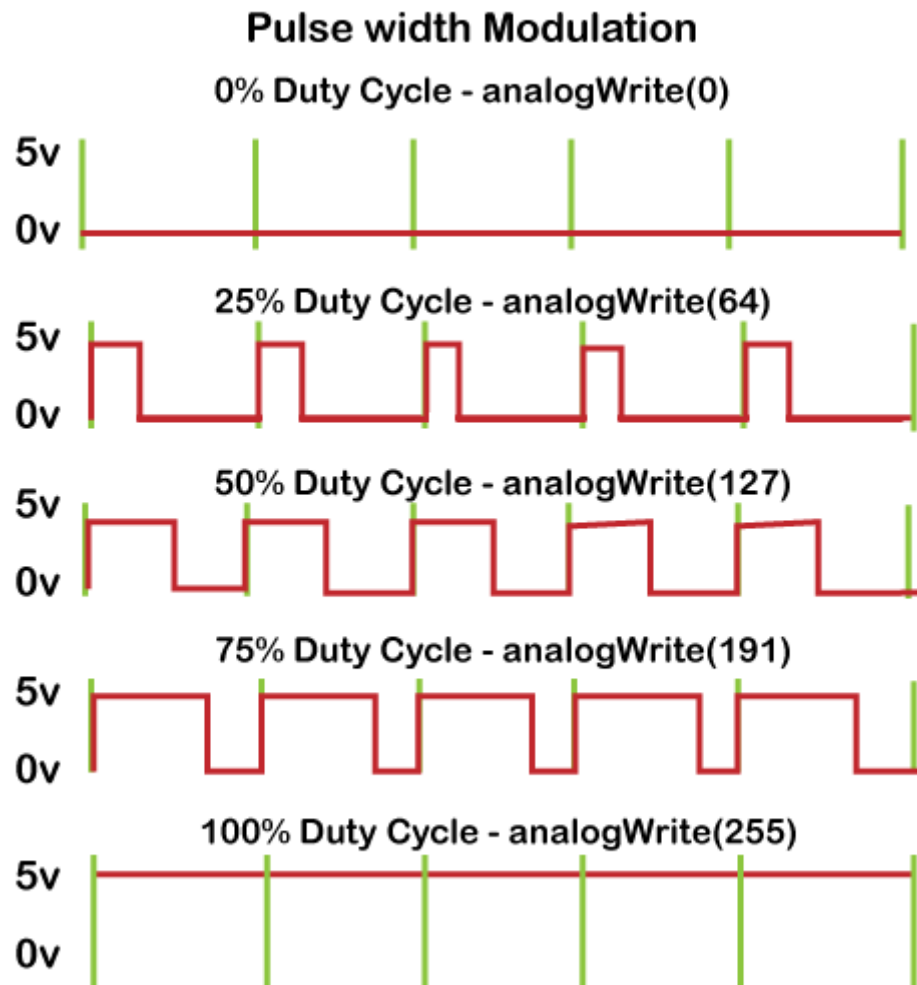
រូបទី៩.១ ការត Volume ជាមួយ Board Arduino UNO

```
int Volume = A5;
void setup( ){
    Serial.begin( 9600 );
}
void loop( ){
    int k = analogRead( Volume );
    Serial.println( k );
    delay( 1000 );
}
```

២. Analog Write (Pule Width Modulation)

PWM ជាសញ្ញា Digital ដែលបំប្លែងជា analog និងប្រើសម្រាប់ធ្វើបម្រែបម្រួលថាមពលផ្គត់ផ្គង់ទៅគ្រឿងទទួលក្នុងបំណងណាមួយដូចជាការធ្វើបម្រែបម្រួលល្បឿនម៉ូទ័រ បម្រែបម្រួលពន្លឺ បម្រែបម្រួលមុំ។ល។ PWM មានរៀងជាស៊ីញ៉ាល់ Digital ហើយការធ្វើបម្រែបម្រួលរយៈពេលបិទ និងបើករបស់ជើង Arduino។ រយៈពេលនោះត្រូវបានគេហៅថា Duty Cycle មានសណ្ឋានដូចខាងក្រោម ៖

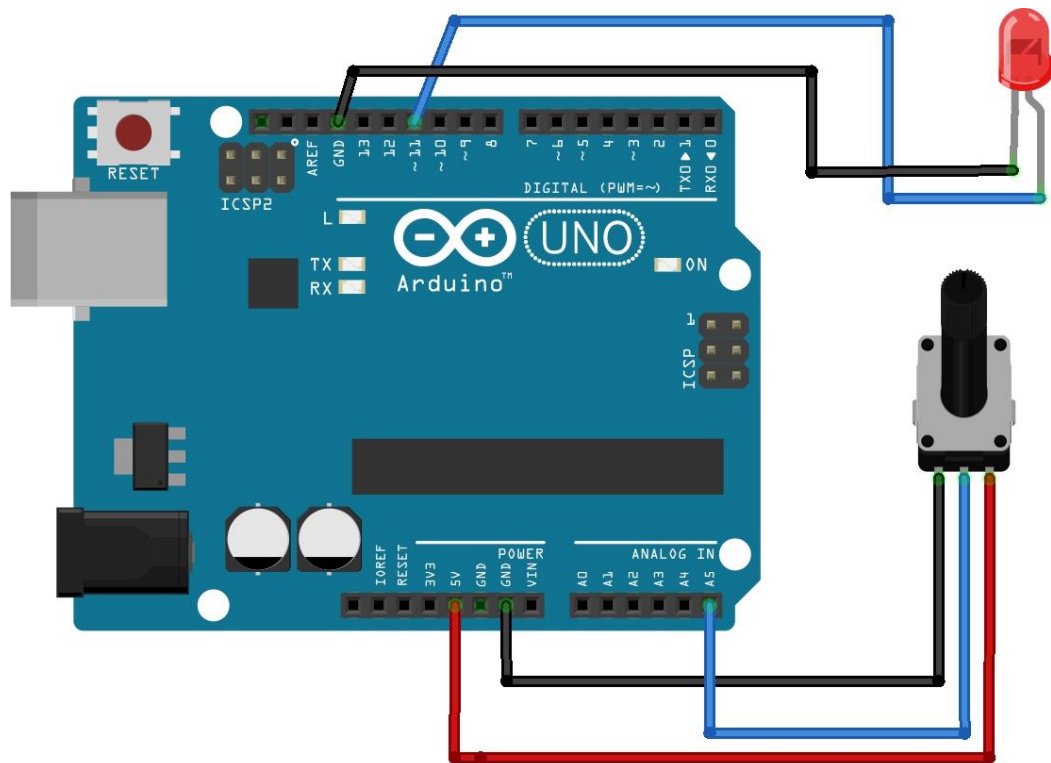
បើរយៈពេលបើកកាន់តែយូរនោះថាមពលផ្គត់ផ្គង់កាន់តែច្រើន ហើយផ្ទុយមកវិញថាមពលនឹង ថយចុះបើរយៈពេលនៃការបិទច្រើនជាងបើក។ ភាគច្រើននៃ Arduino PWM ស្ថិតនៅលើជើងទី 3,5,6,9,10,11 ។ សម្រាប់ Arduino Mega ចាប់ពីជើងទី 2-13 និង 44-46។



រូបទី៩.២ បម្រែបម្រួលនៃការធ្វើការរបស់ជើង Analog Write

ឧទាហរណ៍១ ចូរសរសេរកូដប្រើ volume ដើម្បីបញ្ជា LED ឱ្យភ្លឺ និងរលត់ទៅវិញ។

```
int Volume = A5;
int LED = 11;
void setup( ) {
    pinMode( Volume,INPUT );
    pinMode( LED, OUTPUT );
}
void loop( ) {
    int VR = analogRead( Volume );
    int V = map( VR, 1, 1023, 0, 255 );
    analogWrite( LED,V );
    Serial.println( V );
}
```



រូបទី៩.៣ ការត Volume និង LED ជាមួយ Borad Arduno UNO

ឧទាហរណ៍២ ចូរសរសេរកូដជាមួយជើង PWM ពីខ្សោយទៅខ្លាំង និងពីខ្លាំងមកខ្សោយវិញ ដោយត LED នៅជើងទី 11 របស់ Arduino។

```
void setup( ) {
}

void loop( ) {
    for(int k = 0; k<255; k++){
        analogWrite( 11,k);
        delay( 50);
    }

    for(int k = 255;k > 0; k--){
        analogWrite( 11,k);
        delay( 50);
    }
}
```

ឧទាហរណ៍៣ ចូរសរសេរកូដដោយប្រើ PWM ដើម្បីប្តូរពណ៌ LED ក្រហម បៃតង ខៀវ ហើយ LED ទាំងនោះតប្លល ជើងទី9, 10, 11 របស់ Arduino។

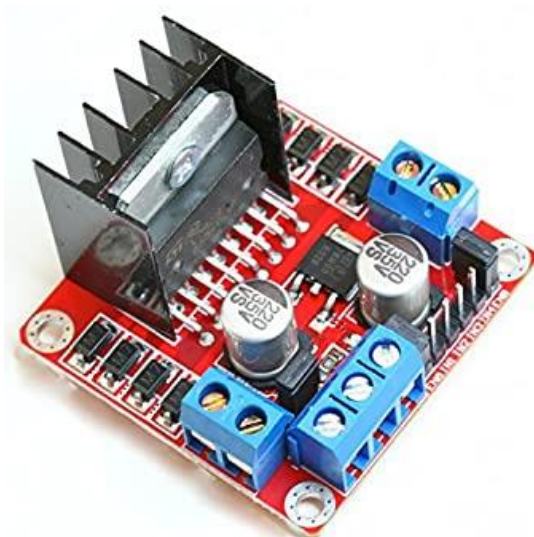
```

int R = 9;
int G = 10;
int B = 11;
void setup( ) {
    pinMode( R, OUTPUT);
    pinMode( G, OUTPUT);
    pinMode( B, OUTPUT);
}
void loop( ) {
    analogWrite( R, 0);
    analogWrite( G, 0);
    analogWrite( B, 0);
    delay( 500 );
    analogWrite( R, 255);
    analogWrite( G, 0);
    analogWrite( B, 0);
    delay( 500 );
    analogWrite( R, 0);
    analogWrite( G, 0255 );
    analogWrite( B, 0 );
    delay( 500 );
    analogWrite( R, 0);
    analogWrite( G, 0);
    analogWrite( B, 255 );
    delay( 500 );
    analogWrite( R, 255);
    analogWrite( G, 255);
    analogWrite( B, 255);
    delay( 500 );
}

```

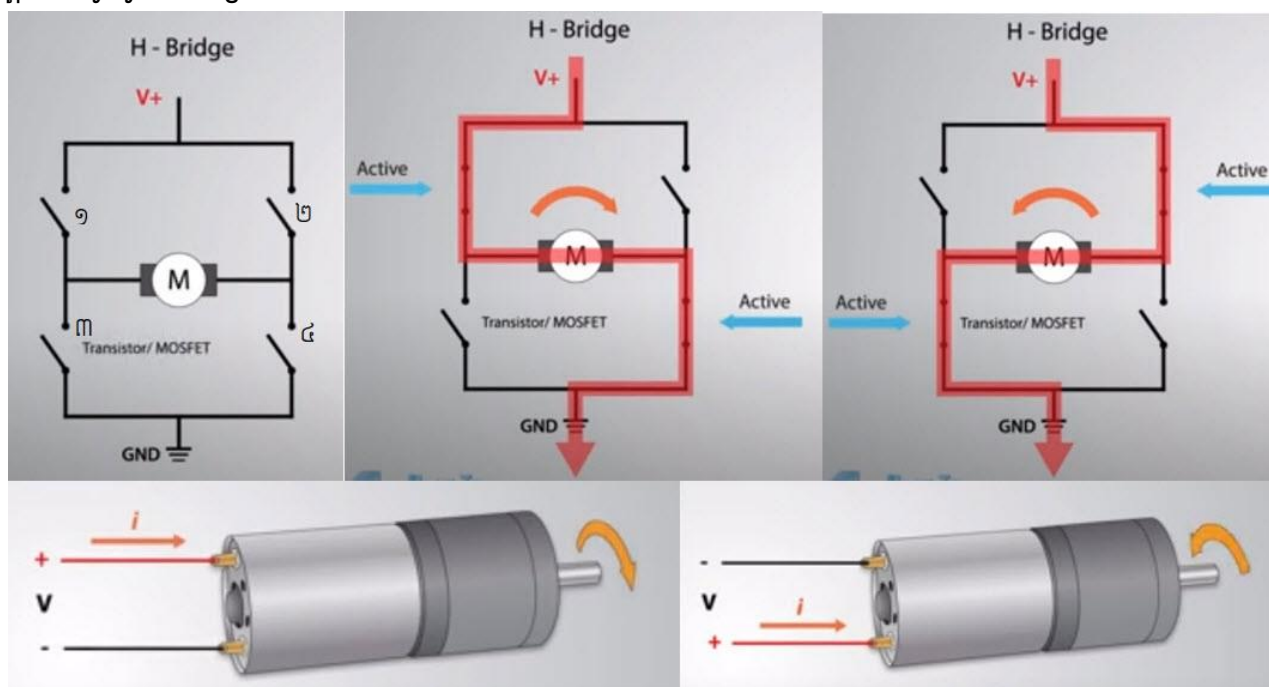
២. DC Motor Control

នៅក្នុងការបញ្ជាម៉ូទ័រចរន្តកាតច្រើន គឺធ្វើការបញ្ជាទៅលើការប្តូរទិសដៅការវិលរបស់ម៉ូទ័រ និងល្បឿនវិលរបស់វា។ ការបញ្ជាទាំងនេះទាមទារឱ្យមានសៀគ្វីអេឡិចត្រូនិចមួយឬហៅថា Drive Motor ក្នុងការសរសេរកូដបញ្ជា។ Drive ទាំងនោះអាចរកទិញបាននៅលើទីផ្សារហើយត្រូវទិញឬដំឡើងអាស្រ័យលើទំហំអនុភាពនៃម៉ូទ័រខាងក្រោម ហើយវាគឺជា Driver ដែលត្រូវបានគេនិយមយកទៅប្រើប្រាស់ក្នុងការធ្វើបូជបញ្ជាផ្សេងៗ។



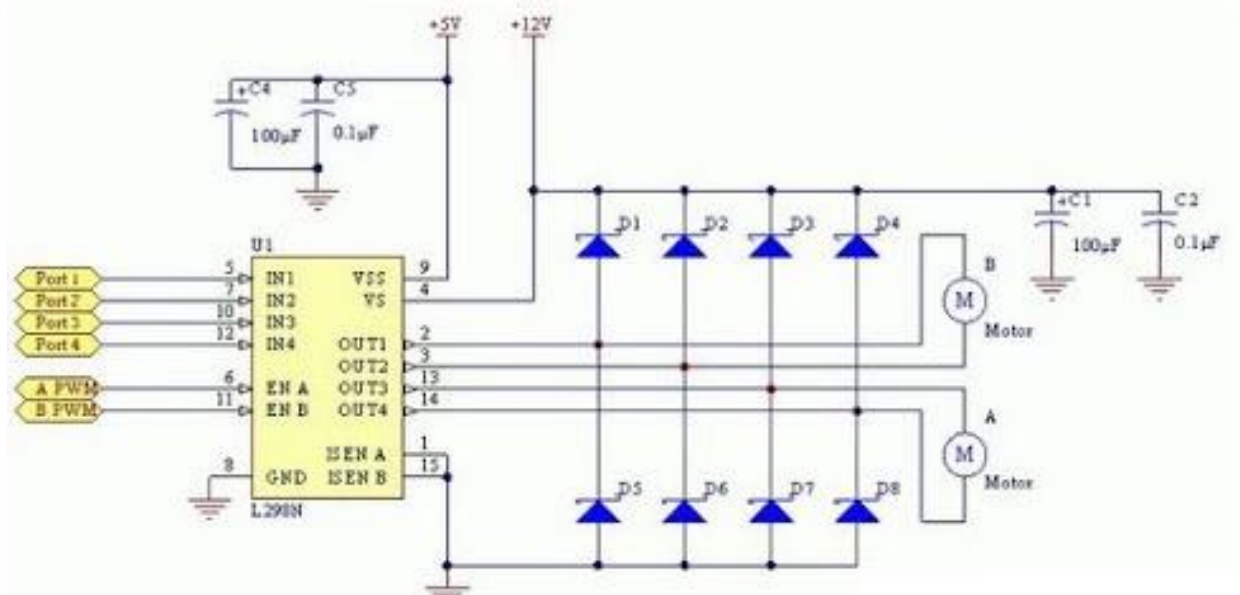
រូបទី៩.៤ L289N motor driver

នៅក្នុង L289N motor driver អាចបញ្ជាម៉ូទ័របាន២ ក្នុងទិសដៅផ្សេងៗគ្នា។ ការធ្វើការរបស់ L289N motor driver ប្រៀបដូចជាកុងតាក់ដោយម៉ូទ័រមួយមានកុងតាក់ចំនួន ៤ ហើយធ្វើការភ្ជាប់កុងតាក់ម្តង២ ដូចរូបខាងក្រោម៖



រូបទី៩.៥ ការធ្វើការរបស់ L289N motor driver

- បើកុងតាក់ទី១ និងកុងតាក់ទី៤ ភ្ជាប់ នោះម៉ូទ័រវិលទៅស្តាំ
- បើកុងតាក់ទី២ និងកុងតាក់ទី៣ ភ្ជាប់ នោះម៉ូទ័រវិលទៅឆ្វេង
- បើកុងតាក់ទី៣ និងកុងតាក់ទី១ ភ្ជាប់ នោះម៉ូទ័រមិនវិល
- បើកុងតាក់ទី៤ និងកុងតាក់ទី២ ភ្ជាប់ នោះម៉ូទ័រមិនវិល



រូបទី៩.៦ សៀគ្វីខាងក្នុងរបស់ L298N motor driver

ឧទាហរណ៍១ ចូរសរសេរកូដបញ្ជាឱ្យម៉ូទ័រទាំងពីរ វិលទៅមុខ។

// Motor A

int IN1 = 2;

int IN2 = 3;

int speedPinA = 6;

// Motor B

int IN3 = 4;

int IN4 = 5;

int speedPinB = 7;

void setup()

{

pinMode(IN1,OUTPUT);

pinMode(IN2,OUTPUT);

pinMode(speedPinA,OUTPUT);

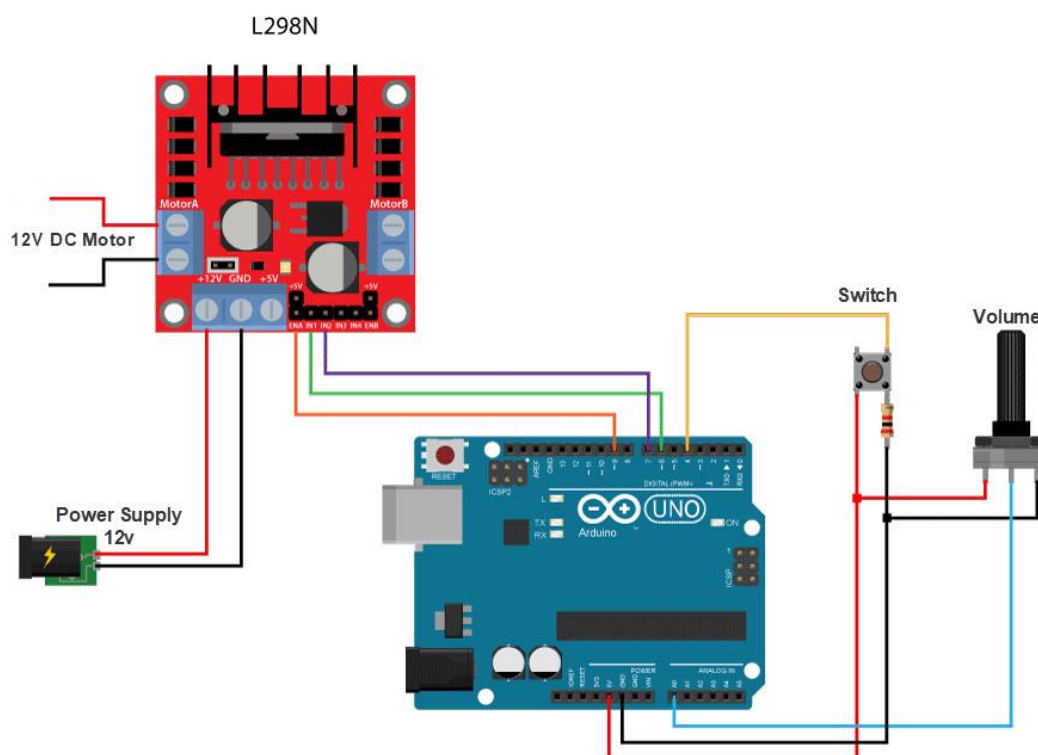
pinMode(IN3,OUTPUT);

```

pinMode( IN4,OUTPUT );
pinMode( speedPinB,OUTPUT );
}
void loop( )
{
    // Motor A
    analogWrite( speedPinA, 255 );
    digitalWrite( IN1, LOW );
    digitalWrite( IN2, HIGH );
    // Motor B
    analogWrite( speedPinB, 255 );
    digitalWrite( IN3, LOW );
    digitalWrite( IN4, HIGH );
}

```

ឧទាហរណ៍២ ចូរសរសេរកូដបញ្ជាឱ្យម៉ូទ័រទៅមុខ ឬថយក្រោយដោយប្រើកុងតាក់ជាអ្នកកំណត់ទិសដៅម៉ូទ័រ ហើយម៉ូទ័រវិលខ្សោយឬខ្លាំង ដោយប្រើ volume ជាអ្នកកំណត់ល្បឿន។



រូបទី៩.៧ សៀគ្វីប្រើក្នុងទី២

```

#define speedPinA 9
#define IN1 6
#define IN2 7
#define button 4
int rotDirection = 0;
int pressed = false;
void setup( ) {
    pinMode( speedPinA, OUTPUT );
    pinMode( IN1, OUTPUT );
    pinMode( IN2, OUTPUT );
    pinMode( button, INPUT );
    // Set initial rotation direction
    digitalWrite( IN1, LOW );
    digitalWrite( IN2, HIGH );
}
void loop( ) {
    int potValue = analogRead( A0 );           // Read potentiometer value
    int pwmOutput = map( potValue, 0, 1023, 0, 255 ); // Map the potentiometer
                                                    // value from 0 to 255
    analogWrite( speedPinA, pwmOutput ); // Send PWM signal to L298N Enable pin
    // Read button - Debounce
    if ( digitalRead( button ) == true ) {
        pressed = !pressed;
    }
    while ( digitalRead( button ) == true );
    delay( 20 );
    // If button is pressed - change rotation direction
    if ( pressed == true & rotDirection == 0 ) {
        digitalWrite( IN1, HIGH );
        digitalWrite( IN2, LOW );
        rotDirection = 1;
        delay( 20 );
    } // If button is pressed - change rotation direction
}

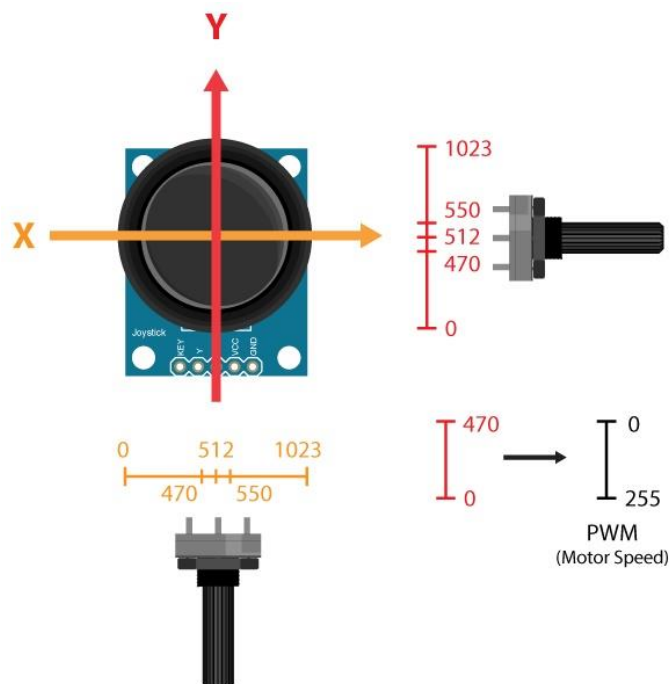
```

```

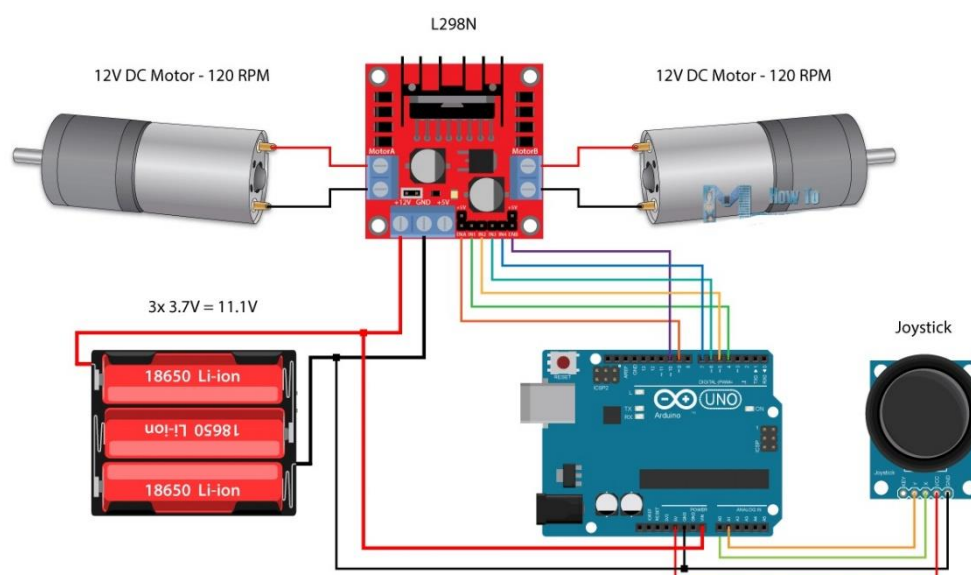
if ( pressed == false & rotDirection == 1 ) {
    digitalWrite( IN1, LOW );
    digitalWrite( IN2, HIGH );
    rotDirection = 0;
    delay( 20 ); }
}

```

ឧទាហរណ៍៣ ចូរបង្កើតកូនឡានដែលបញ្ជាដោយ joystick។ ឧបករណ៍ដែលត្រូវប្រើមាន joystick, ម៉ូទ័រ 12V ចំនួន ២គ្រាប់, ថ្ន 3.7V ចំនួន ៣គ្រាប់ ស្មើនឹង 11V, L289N motor driver និង Board Arduino UNO ។ យើងកំណត់យក 470 ដល់ 550 ជាចំណុចកណ្តាល ដែលអាចកំណត់ឱ្យម៉ូទ័រទាំងពីររើសទៅខាងណា។



រូបទី៩.៨ Joystick



រូបទី៩.៩ ការតភ្ជាប់គ្នាប្រើក្នុងឧទាហរណ៍ទី៣

```

#define speedPinA 9
#define IN1 4
#define IN2 5
#define speedPinB 10
#define IN3 6
#define IN4 7
int motorSpeedA = 0;
int motorSpeedB = 0;
void setup( ) {
    pinMode( speedPinA, OUTPUT );
    pinMode( speedPinB, OUTPUT );
    pinMode( IN1, OUTPUT );
    pinMode( IN2, OUTPUT );
    pinMode( IN3, OUTPUT );
    pinMode( IN4, OUTPUT );
}
void loop( ) {
    int xAxis = analogRead( A0 );    // Read Joysticks X-axis
    int yAxis = analogRead( A1 );    // Read Joysticks Y-axis
    // Y-axis used for forward and backward control
    if ( yAxis < 470 ) {
        // Set Motor A backward
        digitalWrite( IN1, HIGH );
        digitalWrite( IN2, LOW );
        // Set Motor B backward
        digitalWrite( IN3, HIGH );
        digitalWrite( IN4, LOW );
        // Convert the declining Y-axis readings for going backward from 470 to
        0 into 0 to 255 value for the PWM signal for increasing the motor speed
        motorSpeedA = map( yAxis, 470, 0, 0, 255 );
        motorSpeedB = map( yAxis, 470, 0, 0, 255 );
    }
    else if ( yAxis > 550 ) {

```

```

        // Set Motor A forward
        digitalWrite( IN1, LOW );
        digitalWrite( IN2, HIGH );

        // Set Motor B forward
        digitalWrite( IN3, LOW );
        digitalWrite( IN4, HIGH );

        // Convert the increasing Y-axis readings for going forward from 550 to
        //1023 into 0 to 255 value for the PWM signal for increasing the motor speed
        motorSpeedA = map( yAxis, 550, 1023, 0, 255 );
        motorSpeedB = map( yAxis, 550, 1023, 0, 255 );
    }

    // If joystick stays in middle the motors are not moving
else {
    motorSpeedA = 0;
    motorSpeedB = 0;
}

    // X-axis used for left and right control
    if ( xAxis < 470 ) {
// Convert the declining X-axis readings from 470 to 0 into increasing 0 to 255 value
        int xMapped = map( xAxis, 470, 0, 0, 255 );

        // Move to left - decrease left motor speed, increase right motor speed
        motorSpeedA = motorSpeedA - xMapped;
        motorSpeedB = motorSpeedB + xMapped;

        // Confine the range from 0 to 255
        if ( motorSpeedA < 0 ) {
            motorSpeedA = 0;
        }
        if ( motorSpeedB > 255 ) {
            motorSpeedB = 255;
        }
    }

    if ( xAxis > 550 ) {
        // Convert the increasing X-axis readings from 550 to 1023 into 0 to 255 value

```

```

    int xMapped = map( xAxis, 550, 1023, 0, 255 );
    // Move right - decrease right motor speed, increase left motor speed
    motorSpeedA = motorSpeedA + xMapped;
    motorSpeedB = motorSpeedB - xMapped;
    // Confine the range from 0 to 255
    if ( motorSpeedA > 255 ) {
        motorSpeedA = 255;
    }
    if ( motorSpeedB < 0 ) {
        motorSpeedB = 0;
    }
}

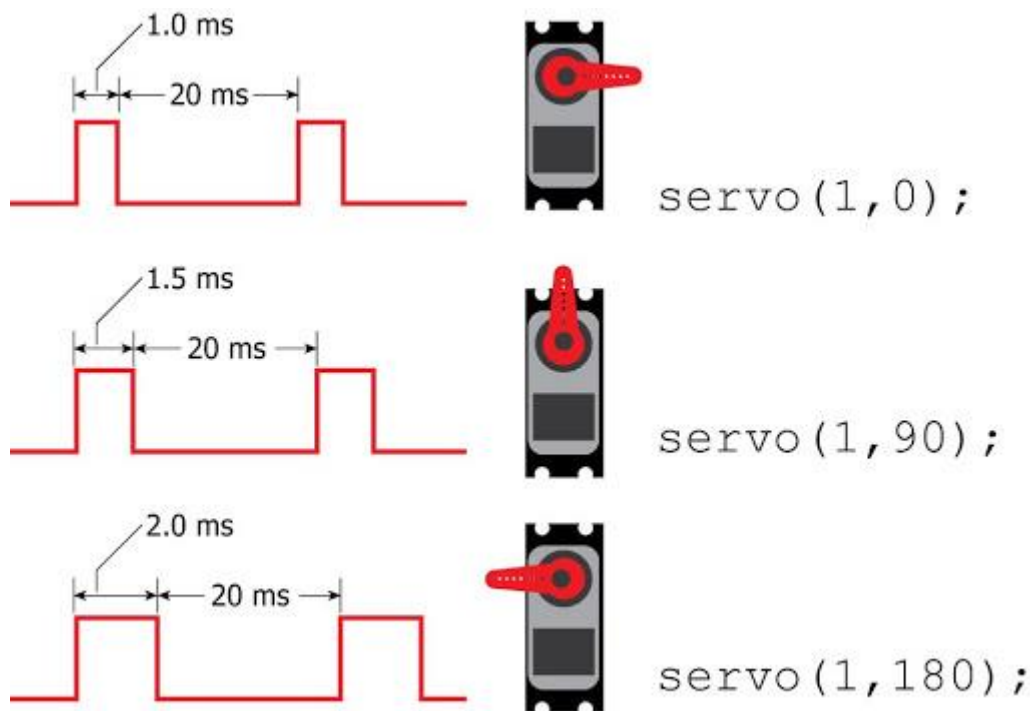
// Prevent buzzing at low speeds ( Adjust according to your motors. My motors
// couldn't start moving if PWM value was below value of 70 )
if ( motorSpeedA < 70 ) {
    motorSpeedA = 0;
}
if ( motorSpeedB < 70 ) {
    motorSpeedB = 0;
}

analogWrite( speedPinA, motorSpeedA );    // Send PWM signal to motor A
analogWrite( speedPinB, motorSpeedB );    // Send PWM signal to motor B
}

```

៣. Servo Motor

Servo Motor ជាប្រភេទម៉ូទ័រដែលយើងអាចបញ្ជាបានតាមរយៈស៊ីញ៉ាល់ឌីជីថល។ ការបញ្ជានេះគឺធ្វើការបញ្ជាទៅកាន់កម្រិតមុំណាមួយដែលយើងចង់បាន។ ជាធម្មតាម៉ូទ័ររបស់វាមាន ចន្លោះពី -៦០ទៅ ៦០ ដឺក្រេ និងពី -៩០ទៅ ៩០ដឺក្រេ អាស្រ័យលើប្រភេទម៉ូទ័រដែលអ្នកទិញ។ Arduino បញ្ជូនស៊ីញ៉ាល់ PWM បញ្ជាប្តូរកម្រិតមុំនោះដែល PWM នោះមាន Duty cycle ប្រហែលពី 0.50ms-2.3ms និងហ្វេកង់ 50Hz។ ខាងក្រោមជាក្រាបបង្ហាញពីស៊ីញ៉ាល់៖



រូបទី៩.១០ ស៊ីញ៉ាល់របស់ Servo Motor

ដើម្បីអាចប្រើ Servo Motor ចាំបាច់ត្រូវមាន Library សម្រាប់ប្រើជាមួយម៉ូឌុលប្រភេទនេះបាន។ យើងអាចទាញយក Library មកប្រើបានតាមរយៈ <https://www.arduino-libraries.info/libraries/servo> ។ ជាទូទៅយើងអាចបញ្ជា Servo Motor បានចំនួន ១២ ម៉ូឌុល សម្រាប់ Board Arduino ធម្មតា និង ៤៨ ម៉ូឌុល សម្រាប់ Board Arduino Mega ។ ចំពោះ Board Arduino ធម្មតា ការប្រើប្រាស់នូវ Library នេះនាំឱ្យមាន ការបិទនូវ PWM ស៊ីញ៉ាល់ដែលត្រូវចេញនៅជើងទី៩ និងជើងទី១០ ទោះជើងទាំងអស់ត្រូវប្រើ ឬមិនប្រើក៏ ដោយ។ ចំណែក Board Arduino Mega ក្នុងករណីប្រើ Servo Motor ក្នុងចំនួនតិចជាង ១២ ម៉ូឌុល ក៏ដោយ ជើង PWM ទាំងអស់នៅដំណើរការដូចធម្មតា លើកលែងតែចាប់ពីចន្លោះ ១២ ទៅ ២៣ ម៉ូឌុលឡើង ជើង PWM នៅជើងទី១១ និងជើងទី១២ ត្រូវបានបិទមិនឱ្យដំណើរការ។

ទម្រង់ដែលប្រើជាមួយ Servo Motor មាន៖

- `attach( )`

`servo.attach( pin )`

`servo.attach( pin, min, max )`

- `write( )`

`servo.write( angle )`

angle: មុំដែលត្រូវវិលពី 0-180

- `writeMicroseconds( )`

`servo.writeMicroseconds( uS )`

uS: គិតជាមីក្រូវិនាទី

- `read( )`

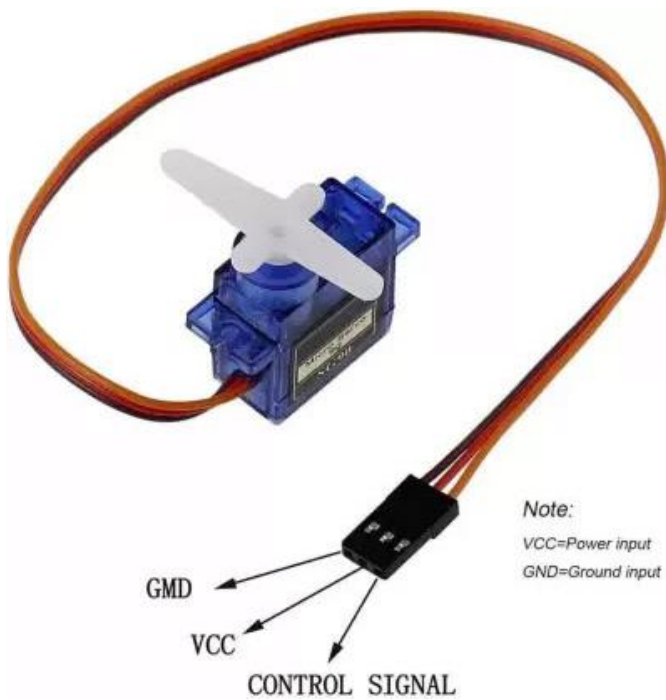
`servo.read( )` ម៉ូដែលម៉ូទ័រស្ថិតនៅ

- `attached( )`

`servo.attached( )` សម្រាប់ពិនិត្យមើលថា តើ servo pin ត្រូវបានប្រកាសឱ្យប្រើឬនៅ

- `detach( )`

`servo.detach( )` ផ្តាច់ជើងនោះពី servo ហើយក្លាយជា Digital ធម្មតាវិញ



រូបទី៩.១១ ស៊ីញ៉ាល់របស់ Servo Motor

Servo Motor មានជើងទាំងអស់ចំនួន៣៖

- ១ ជើង Signal
- ២ ជើង VCC ភ្លើងចូល
- ៣ ជើង GND ម៉ាស

គំរូកូដមួយសម្រាប់តេស្ត Servo Motor

```
#include<Servo.h>

Servo myservo;
int pos = 0;

void setup( ) {
    myservo.attach( 9 );
}

void loop( ) {
    for ( pos = 0; pos <= 180; pos += 1 ) {
        myservo.write( pos );
        delay( 15 );
    }
}
```

```

    for ( pos = 180; pos >= 0; pos -= 1 ) {
        delay( 15 );
    }
}

```

ឧទាហរណ៍១ ការប្រើ Volume គ្រប់គ្រងទិសដៅរបស់ Servo Motor។

```

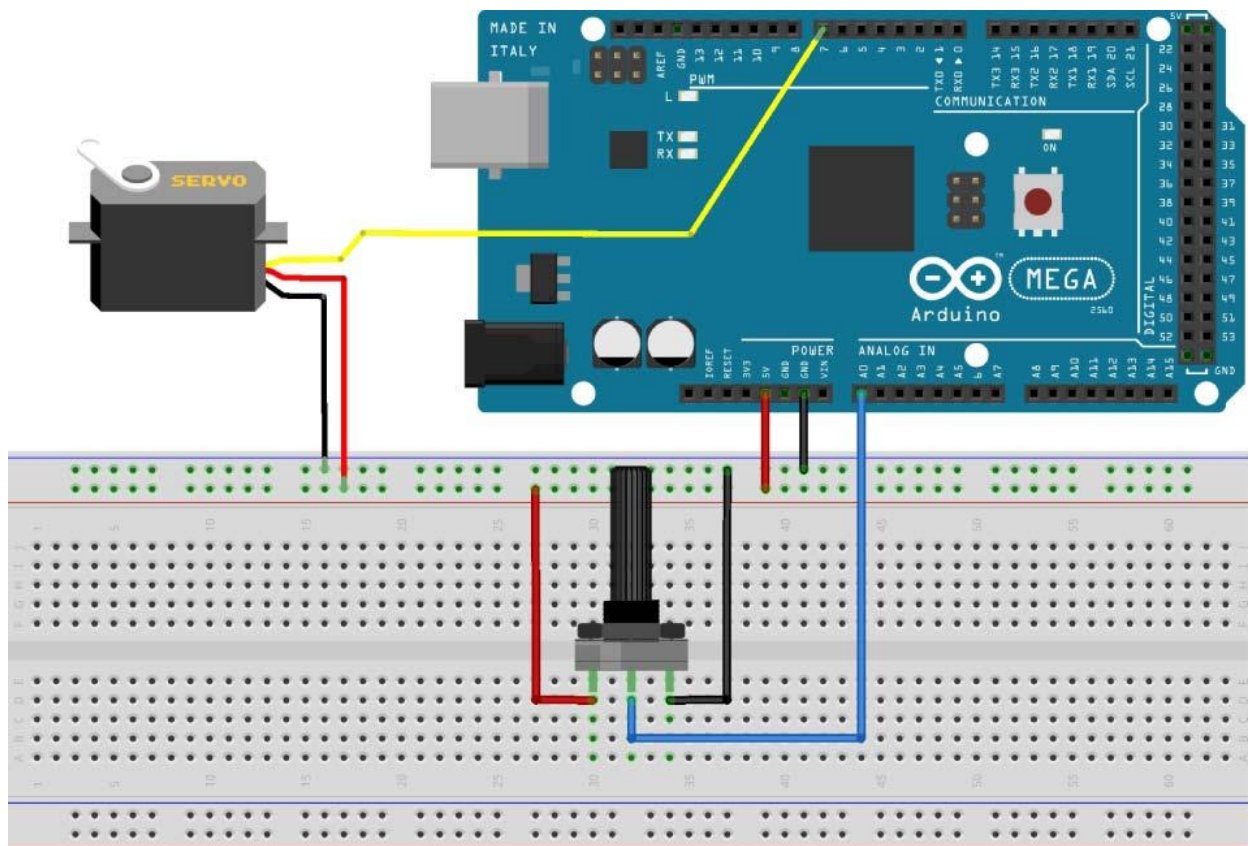
#include <Servo.h>

Servo myServo;

void setup( ) {
    myServo.attach( 7 );
}

void loop( ) {
    int potValue = analogRead( A0 );
    int angleValue = map( potValue, 0, 1023, 0, 180 );
    myServo.write( angleValue );
    delay( 10 );
}

```



រូបទី៩.១២ ការតភ្ជាប់រវាង Servo Motor + Volume និង Board Arduino Mega ប្រើក្នុងឧទាហរណ៍ទី១

មេរៀនទី១០ Sensor

១. Thermistor

Thermistor វាជាអេស៊ីស្តង់អគ្គិសនីដែលតម្លៃរបស់វាប្រែប្រួលខ្លាំងតាមរយៈកម្ដៅពីខាងក្រៅ ។ វាត្រូវបានគេប្រើសម្រាប់រង្វាស់សីតុណ្ហភាពក្នុងកម្រិតសីតុណ្ហភាពខ្ពស់ល្មម ល្បឿនបន្តិចបាត់ពីចន្លោះ -90c – 300c។ Thermistor មានភាពខុសគ្នាពី Resistance Temperature Detectors (RTDs) ដោយសារធាតុផ្សំប្រើក្នុងវាជាធម្មតាធ្វើពីសេរ៉ាមិច ឬ Polymer ខណៈដែលធាតុផ្សំ RTDs ធ្វើពីលោហៈសុទ្ធហើយអាចវាស់រហូតដល់រាប់រយពាន់អង្សាសេ។ ដោយសារលក្ខណៈពិសេសខាងលើរបស់ Thermistor ត្រូវបានគេចាត់ទុកដូចជា Sensor មួយដែរ។

នៅក្នុងការគណនាសីតុណ្ហភាព គេប្រើសមីការ Steinhart-Hart ដែលមានរូបមន្ត

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{\beta} + \ln \frac{R}{R_0}$$

T សីតុណ្ហភាពដែលត្រូវវាស់ គិតជាកែវីន ()

T₀ សីតុណ្ហភាពនៅបន្ទប់ពិសោធន៍ 25 អង្សា

ស្មើ 298.15k

β ចំនួនថេរ NTC3950 មានន័យថា β=3950

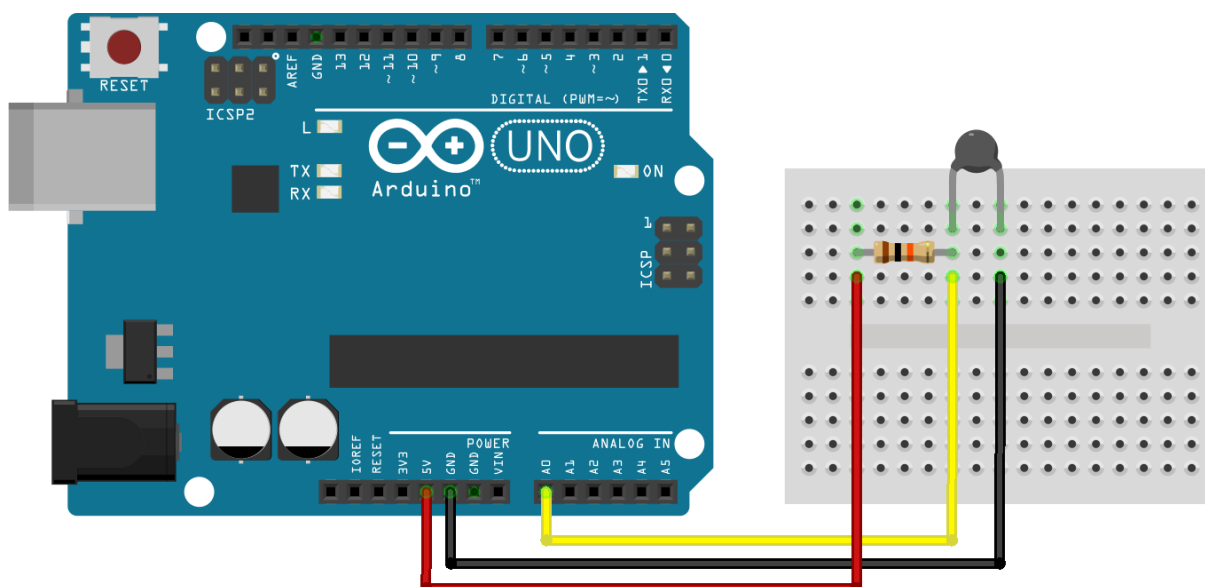
R អេស៊ីស្តង់នៅសីតុណ្ហភាពកំពុងវាស់

R₀ អេស៊ីស្តង់នៅបន្ទប់ពិសោធន៍ក្នុងសីតុណ្ហភាព

25 អង្សា ស្មើ 10Kohm



រូបទី១០.១ Thermistor



រូបទី១០.២ ការតភ្ជាប់ Thermistor ជាមួយ Board Arduino Uno

```

កូដគំរូរបស់ Thermistor
const float T0=298.15;
const float R0=10000;
const float belta=3950;
const float r = 10000;
float val;
float R,vol,T;
void setup( ) {
    Serial.begin( 9600 );
}
void loop( ) {
    val=analogRead( A0 );
    vol=( val * 5 )/1023;
    R=( r * ( 5-vol ) )/vol;
    T = ( 0 * belta )/ ( belta+( T0 * log( R/R0 ) ) );
    T = T-273;
    Serial.println( String( "Temperatur=" )+String( T ) );
    delay( 500 );
}

```

ឧទាហរណ៍ទី១ ចូរសរសេរកូដឱ្យបង្ហាញសីតុណ្ហភាពនៅលើ Serial Monitor។

```

int ThermistorPin = 0;
int Vo;
float R1 = 1000; // Resistor Value ( We use 1 KOhms Resistor )
float logR2, R2, T, Tc, Tf;
float A = 1.009249522e-03, B = 2.378405444e-04, C = 2.019202697e-07;
void setup( ) {
    Serial.begin( 9600 );
}
void loop( ) {
    Vo = analogRead( ThermistorPin );
    R2 = R1 * ( 1023.0 / ( float)Vo - 1.0 );
    logR2 = log( R2 );

```

$$T = (1.0 / (A + B * \log R2 + C * \log R2 * \log R2 * \log R2));$$

$$T_c = T - 273.15;$$

$$T_f = (T_c * 9.0) / 5.0 + 32.0;$$

```
Serial.print( "Temperature: " );
```

```
Serial.print( Tc );
```

```
Serial.print( " C; " );
```

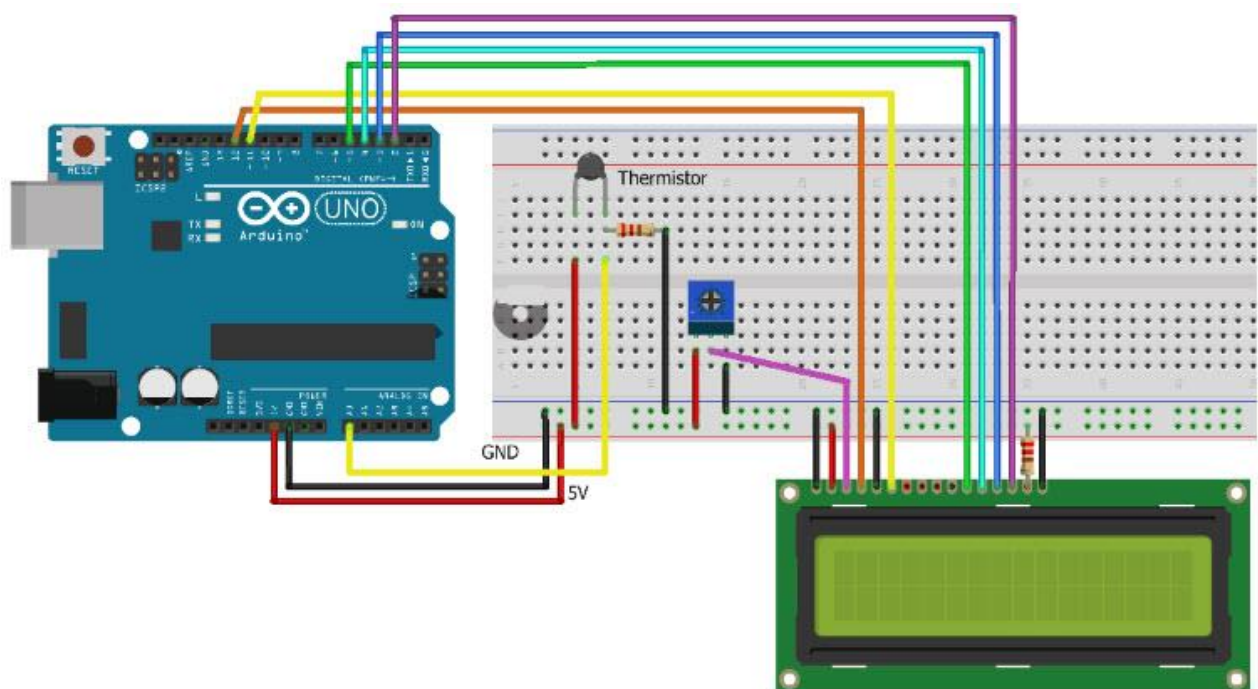
```
Serial.print( Tf );
```

```
Serial.println( " F" );
```

```
delay( 500 );
```

```
}
```

ឧទាហរណ៍ទី២ ចូរសរសេរកូដឱ្យបង្ហាញសីតុណ្ហភាពនៅលើ LCD 16x2។



រូបទី១០.៣ ការតភ្ជាប់ Thermistor និង LCD ជាមួយ Board Arduino Uno

/*

The circuit: ការតភ្ជាប់

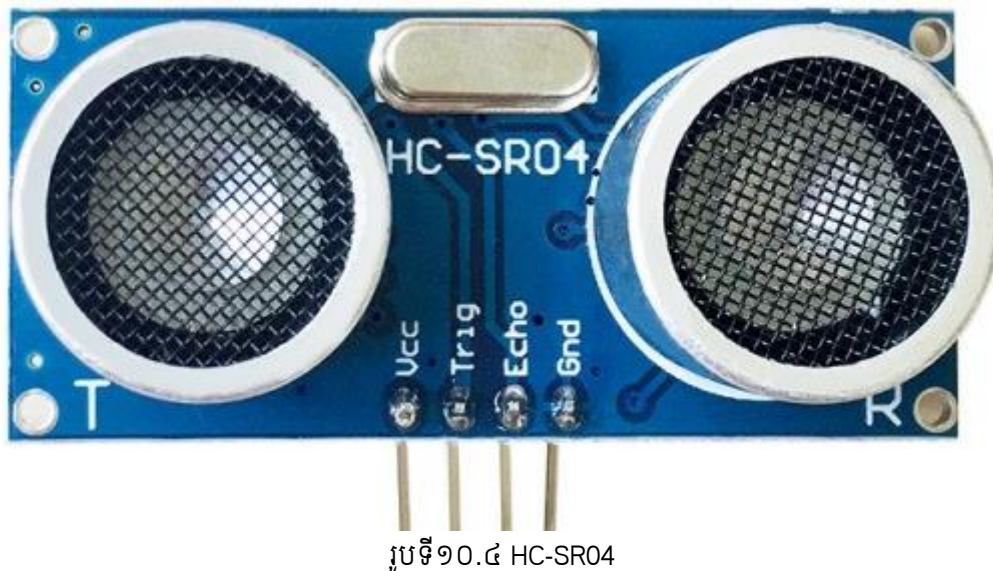
- * LCD RS pin to digital pin 12
- * LCD Enable pin to digital pin 11
- * LCD D4 pin to digital pin 5
- * LCD D5 pin to digital pin 4
- * LCD D6 pin to digital pin 3

- * LCD D7 pin to digital pin 2
- * LCD R/W pin to ground
- * LCD VSS pin to ground
- * LCD VCC pin to 5V
- * 10K resistor:
- * ends to +5V and ground
- * wiper to LCD VO pin (pin 3)
- */

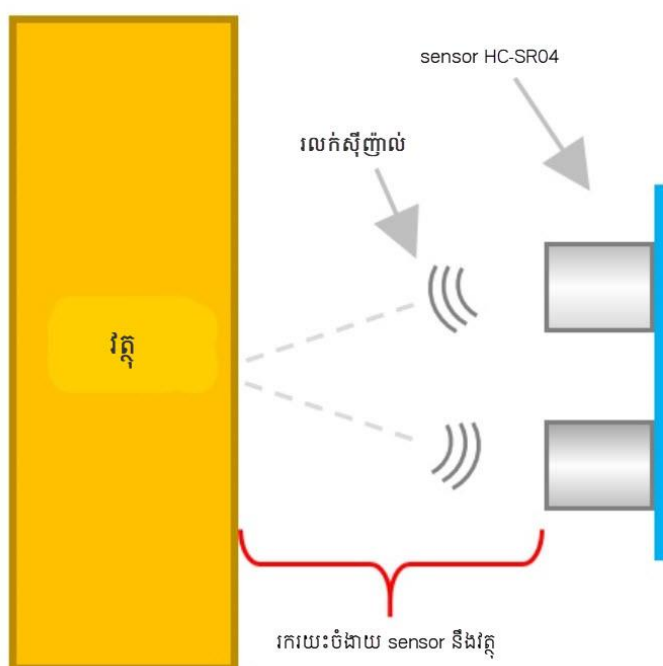
```
#include <LiquidCrystal.h>
LiquidCrystal lcd( 12, 11, 5, 4, 3, 2 );
int ThermistorPin = 0;
int Vo;
float R1 = 1000; // Resistor Value ( We use 1 KOhms Resistor )
float logR2, R2, T, Tc, Tf;
float A = 1.009249522e-03, B = 2.378405444e-04, C = 2.019202697e-07;
void setup( ) {
  lcd.begin( 16, 2 );
}
void loop( ) {
  Vo = analogRead( ThermistorPin );
  R2 = R1 * ( 1023.0 / ( float )Vo - 1.0 );
  logR2 = log( R2 );
  T = ( 1.0 / ( A + B*logR2 + C*logR2*logR2*logR2 ) );
  Tc = T - 273.15;
  Tf = ( Tc * 9.0 )/ 5.0 + 32.0;
  lcd.print( "Temp: " );
  lcd.setCursor( 0,1 );
  lcd.print( Tc );
  lcd.print( " C; " );
  lcd.print( Tf );
  lcd.print( " F" );
  delay( 500 );
}
```

២. Distance Sensor

យើងប្រើ Ultrasonic sensor module HC-SR04 សម្រាប់វាស់ចម្ងាយ។ Ultra sound ដែលបានផ្លាស់ទី 340m/s និងទៅប៉ះជាមួយវត្ថុ ហើយឡុងត្រលប់មកវិញក្នុងរយៈពេលមួយ។ យើងធ្វើការវាស់ រយៈពេលនោះរួចគណនាចេញជាចម្ងាយរៀបជាមួយវត្ថុនោះ។



គោលការណ៍សំខាន់ក្នុងការវាស់រយៈចម្ងាយរបស់ HC-SR04 គឺការបញ្ជូនលេកស៊ីញ៉ាល់សម្លេង Ultrasonic ចេញទៅតាម Transmitter នៅពេលដែលលេកស៊ីញ៉ាល់សម្លេងទៅបុកនឹងវត្ថុអ្វីមួយ នោះលេកស៊ីញ៉ាល់សម្លេងឡុងត្រលប់មកវិញតាម Receiver ។ ដោយចាប់រយៈពេល តាំងពីពេលចេញ លេកស៊ីញ៉ាល់សម្លេងទៅរហូតមកដល់វិញ ដែលធ្វើឱ្យយើងអាចរករយៈចម្ងាយពី sensor ទៅវត្ថុនោះបាន។



រូបទី១០.៥ ការធ្វើការរបស់ sensor HC-SR04

រូបមន្តក្នុងការគណនារកចម្ងាយ

$$\text{Time} = \text{distance} / \text{speed}$$

$$\text{រយៈពេល} = \text{រយៈចម្ងាយ} / \text{ល្បឿនសម្លេង}$$

$$t = s / v \text{ ឬ}$$

$$\text{រយៈចម្ងាយ} = \text{រយៈពេល} * \text{ល្បឿនសម្លេង}$$

$$s = t * v$$

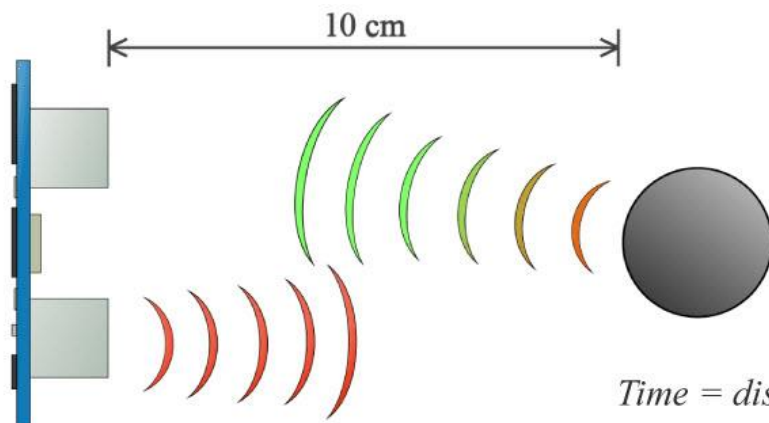
យើងដឹងថាល្បឿនសម្លេងស្មើ 340m/s

ឬស្មើ 0.034cm/us ។ ដោយល្បឿន

នេះ ជាល្បឿនទាំងទៅទាំងមក ដូចនេះ យើងត្រូវចែកនឹង 2 ដើម្បីបានចម្ងាយពី sensor ទៅវត្ថុ

$$s = t * 0.034 / 2$$

ឧទាហរណ៍១ របៀបគណនា រករយៈពេល



speed of sound:

$$v = 340 \text{ m/s}$$

$$v = 0,034 \text{ cm}/\mu\text{s}$$

Time = distance / speed:

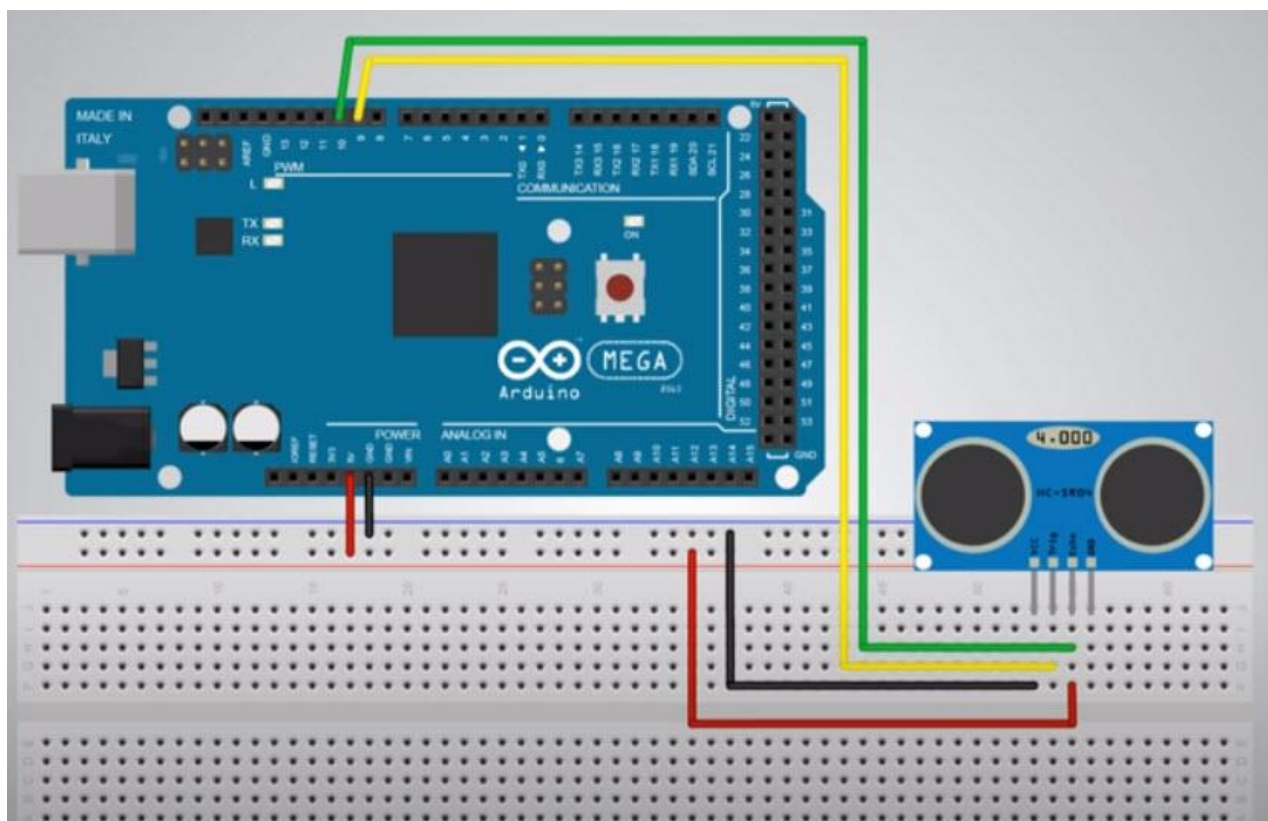
$$t = s / v = 10 / 0,034 = 294 \mu\text{s}$$

Distance:

$$s = t \cdot 0,034 / 2$$

រូបទី១០.៦ ការបញ្ចេញ រលកស៊ីញ៉ាល់

ឧទាហរណ៍២ ចូរសរសេរកូដដាស់ចម្ងាយវត្ថុ ដោយបង្ហាញប្រព័ន្ធនៅលើ Serial Monitor ។



រូបទី១០.៧ ការភ្ជាប់ sensor HC-SR04 ជាមួយ board arduino Uno

```
const int trigPin = 9;
const int echoPin = 10;

// defines variables
long duration;
```

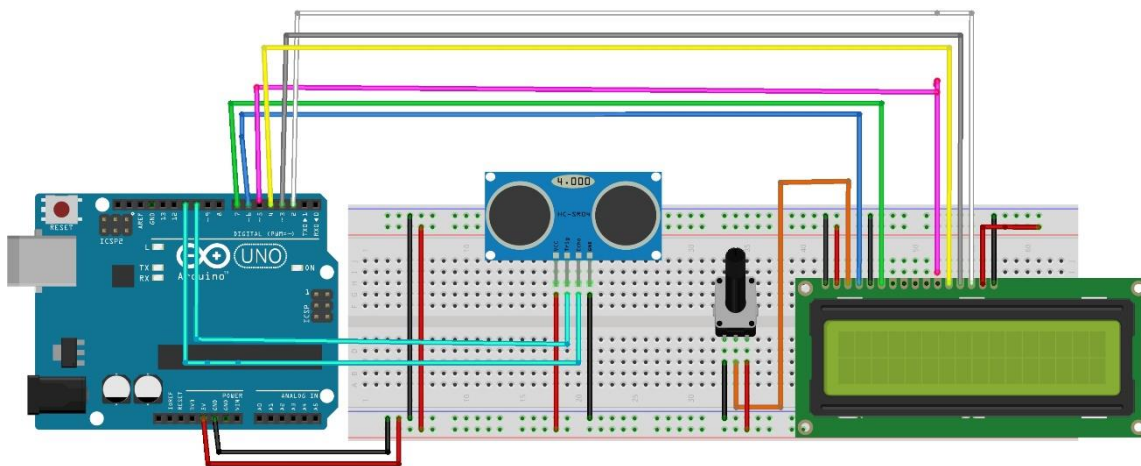
រៀបរៀងដោយលោក គាន ប៊ុនធា

```

int distance;
void setup( ) {
  pinMode( trigPin, OUTPUT ); // Sets the trigPin as an Output
  pinMode( echoPin, INPUT ); // Sets the echoPin as an Input
  Serial.begin( 9600 ); // Starts the serial communication
}
void loop( ) {
  // Clears the trigPin
  digitalWrite( trigPin, LOW );
  delayMicroseconds( 2 );
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite( trigPin, HIGH );
  delayMicroseconds( 10 );
  digitalWrite( trigPin, LOW );
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn( echoPin, HIGH );
  // Calculating the distance
  distance= duration *0.034/2;
  // Prints the distance on the Serial Monitor
  Serial.print( "Distance: " );
  Serial.println( distance ); }

```

ឧទាហរណ៍៣ ចូរសរសេរកូដវាស់ចម្ងាយវត្ថុ ដោយបង្ហាញប្រវែងនៅលើ LCD ដោយប្រវែងគិតជា cm និង inch ។



រូបទី១០.៨ ការភ្ជាប់ sensor HC-SR04 និង LCD ជាមួយ board arduino Uno

```

#include <LiquidCrystal.h>           // includes the LiquidCrystal Library
LiquidCrystal lcd( 6 , 7, 5, 4, 3, 2); // Creates an LCD object

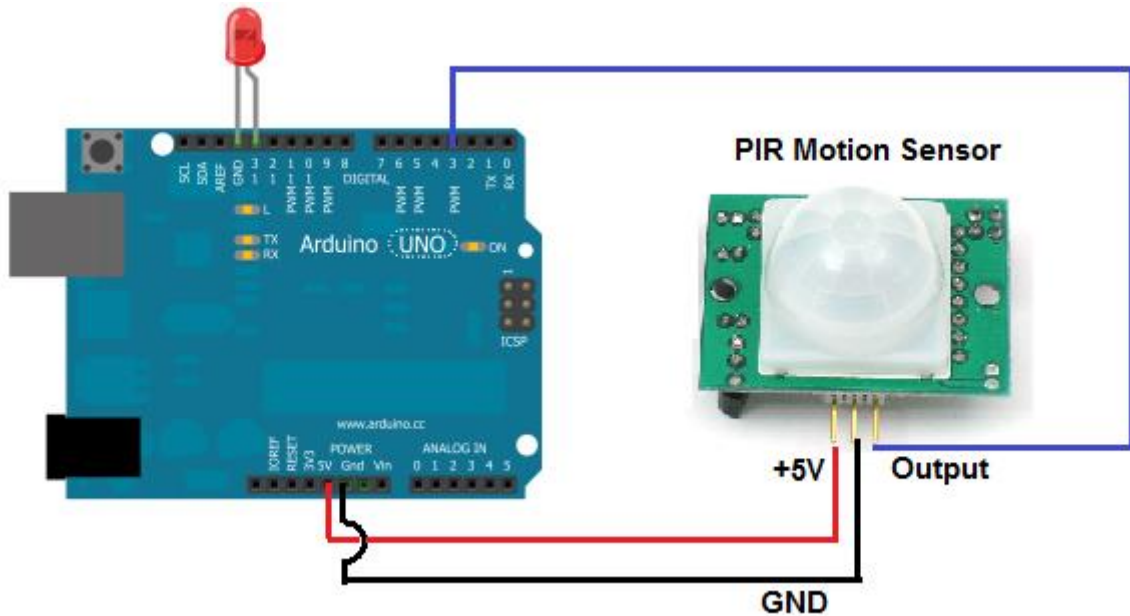
const int trigPin = 11;
const int echoPin = 10;
long duration;
int distanceCm, distanceInch;
void setup( ) {
  lcd.begin( 16,2 ); // Initializes the interface to the LCD screen, and specifies the dimensions
  ( width and height ) of the display
  pinMode( trigPin, OUTPUT );
  pinMode( echoPin, INPUT );
}
void loop( ) {
  digitalWrite( trigPin, LOW );
  delayMicroseconds( 2 );
  digitalWrite( trigPin, HIGH );
  delayMicroseconds( 10 );
  digitalWrite( trigPin, LOW );
  duration = pulseIn( echoPin, HIGH );
  distanceCm= duration *0.034/2;
  distanceInch = duration *0.0133/2;
  lcd.setCursor( 0,0 ); // Sets the location at which subsequent text written to the LCD will be
  displayed
  lcd.print( "Distance: " ); // Prints string "Distance" on the LCD
  lcd.print( distanceCm ); // Prints the distance value from the sensor
  lcd.print( " cm" );
  delay( 10 );
  lcd.setCursor( 0,1 );
  lcd.print( "Distance: " );
  lcd.print( distanceInch );
  lcd.print( " inch" );
  delay( 10 );
}

```

៣. Security Sensor

ក. Motion Sensor

ជាប្រភេទ Sensor ដែលធ្វើការនៅពេលដែលមានវត្ថុ ឬការវាស់វែងជីវិតផ្លាស់ទីមានចលនានៅចំពីមុខ Sensor ។ ជាធម្មតា Motion Sensor ឱ្យ OUTPUT ជាឌីជីថល ចំណែកកូដរបស់វា គឺប្រើ ដើម្បីចាប់យក ស៊ីញ៉ាល់។



រូបទី១០.៩ ការភ្ជាប់ Motion Sensor និង LED ជាមួយ board arduino Uno

ឧទាហរណ៍ទី១ ចូរសរសេរកូដឱ្យបង្ហាញនៅលើ Serial Monitor ថា Motion Sensor ចាប់បានស៊ីញ៉ាល់ ឬ អត់។

```
int PIR = 3;
int val = 0;
void setup( ) {
    pinMode( PIR, INPUT ); // sets the pin as input
    Serial.begin( 9600 );
}
void loop( ) {
    val = digitalRead( PIR );
    Serial.print( "val = " );
    Serial.println( val );
}
```

ឧទាហរណ៍ទី២ ចូរសរសេរកូដបញ្ជាឱ្យអំពូល LED ភ្លឺពេល Motion Sensor ចាប់បានវត្ថុអ្វីមួយ។

```
int led= 13;
int PIR= 3;
void setup( ){
    pinMode( led, OUTPUT );
    pinMode( PIR, INPUT );
}
void loop( ){
    int value= digitalRead( PIR );
    if ( value == HIGH )
    {
        digitalWrite( led, HIGH );
        delay( 1000 ) ;
    }
    else
    {
        digitalWrite( led, LOW );
        delay( 1000 ) ;
    }
}
```

២. Flame Sensor

ជាប្រភេទ Sensor ប្រើសម្រាប់វាស់ ឬក៏ចាប់យកស៊ីញ៉ាល់នៅពេលមានអគ្គិភ័យកើតឡើង។ Flame Sensor ចាប់យកតែពន្លឺអណ្តាតភ្លើង ហើយមិនចាប់យកផ្សែងនោះទេ។ Flame Sensor មាន OUTPUT ពីរប្រភេទគឺ Digital and Analog ដែលមានន័យថា យើងអាចប្រើ DigitalRead ក៏បាន AnalogRead ក៏បាន។

ឧទាហរណ៍ ចូរសរសេរកូដបញ្ជាឱ្យអំពូល LED ភ្លឺនៅពេលដែលមានអគ្គិភ័យកើតឡើង ហើយអំពូលលេចនៅពេលដែលគ្មានអគ្គិភ័យ។

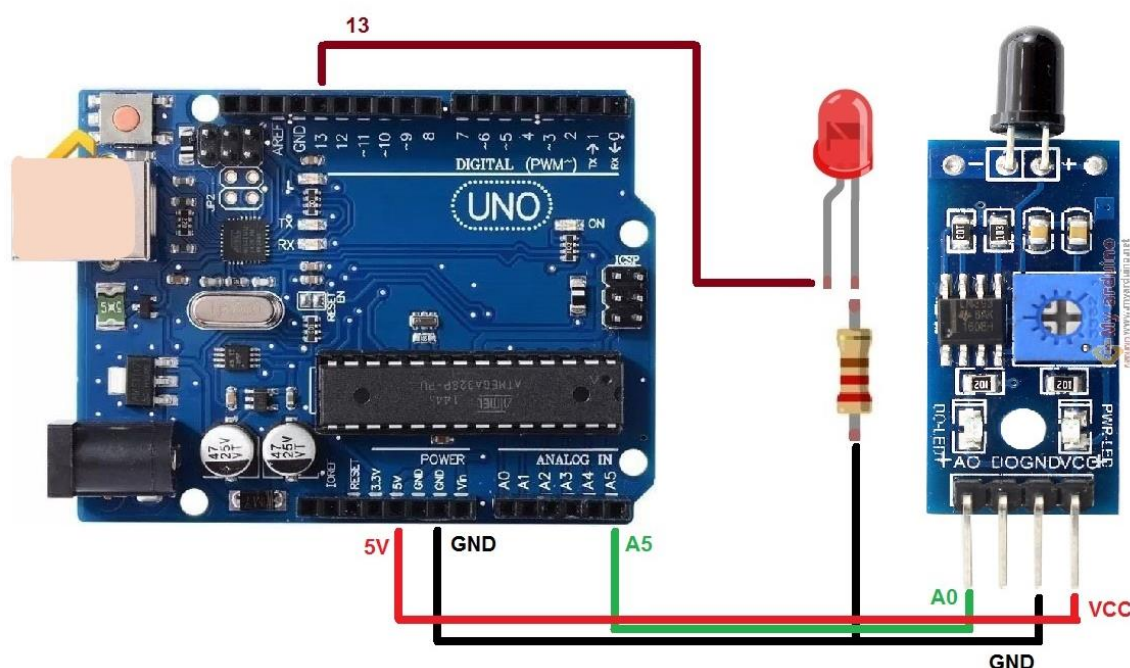
```
int ledPin = 13;
int analogPin = 5;
int val = 0;
```

```

void setup( ) {
    pinMode( ledPin, OUTPUT );
    Serial.begin( 9600 );
}

void loop( ) {
    val = analogRead( analogPin );
    Serial.print( "val = " );
    Serial.println( val );
    if ( val < 900 ) {
        digitalWrite( ledPin, HIGH );
    }
    else {
        digitalWrite( ledPin, LOW );
    }
    delay( 100 );
}

```



រូបទី១០.១០ ការភ័ក្ត្រ Flame Sensor និង LED ជាមួយ board arduino Uno

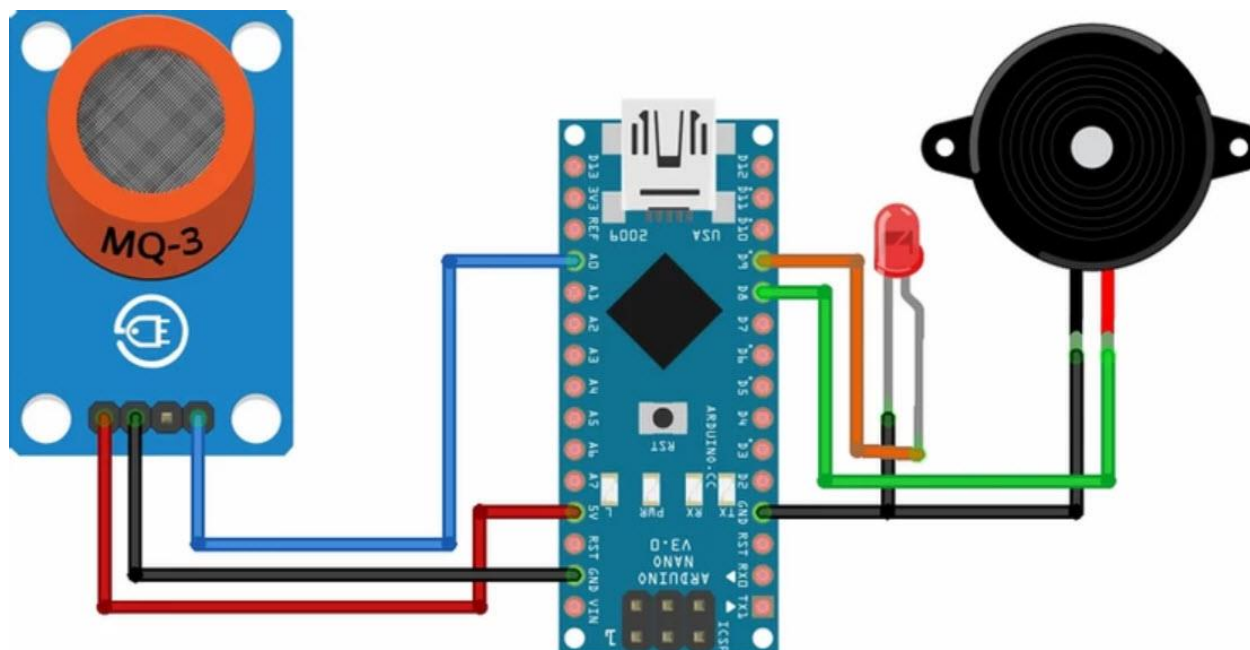
យើងអាចយក Flame Sensor មកធ្វើជាឧបករណ៍ត្រួតចាប់អគ្គិភ័យ ធ្វើជាសញ្ញាប្រកាសអាសន្ន ឬ សម្លេងប្រកាសអាសន្នផ្សេងៗ តាមតម្រូវការ។

គ. Alcohol Sensor

សម្រាប់វាស់កម្រិតជាតិអាកុល ដោយមាន OUTPUT ពីរគឺ Digital and Analog ។

ឧទាហរណ៍ ចូរសរសេរកូដបញ្ជាឱ្យអំពូល LED ភ្លឺ និងមានសម្លេង Buzzer ឮពីដងៗ នៅពេលដែលមានជាតិអាកុលនៅជិត។

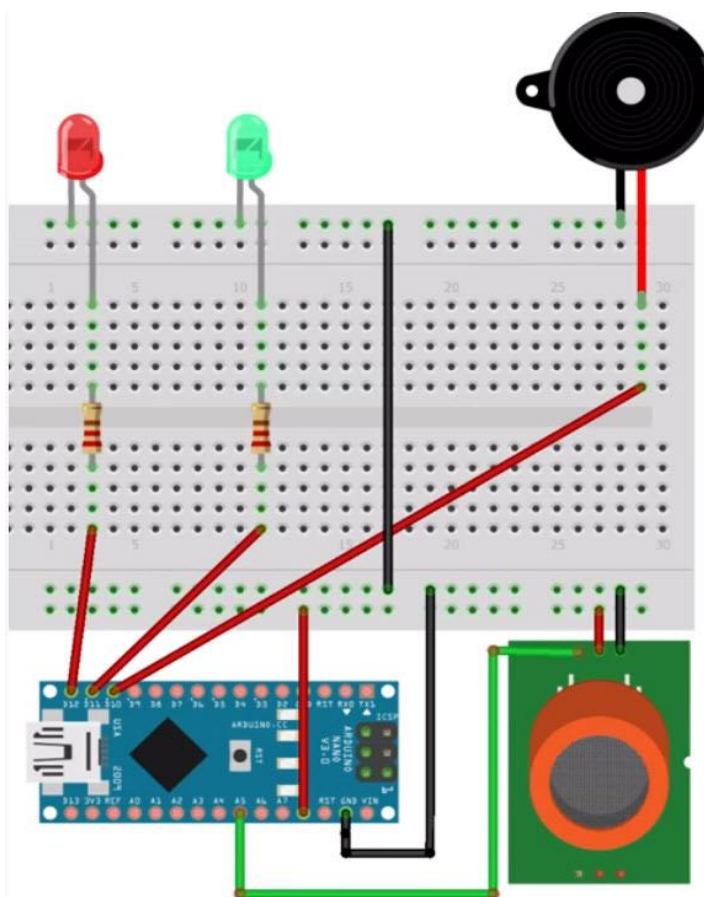
```
const int MQ3 = 0;
const int Buzzer = 8;
const int LED = 9;
int value;
void setup( )
{
    Serial.begin ( 9600 );
    pinMode ( MQ3,INPUT );
    pinMode ( Buzzer, OUTPUT );
    pinMode ( LED, OUTPUT );
    digitalWrite( Buzzer,LOW );
    digitalWrite( LED,LOW );
}
void loop( ) {
    value = analogRead( MQ3 );
    Serial.println( value );
    if( value>440 )
    {
        digitalWrite( Buzzer,HIGH );
        digitalWrite( LED,HIGH );
    }
    else {
        digitalWrite( Buzzer,LOW );
        digitalWrite( LED,LOW );
    }
    delay( 500 );
}
```



រូបទី១០.១១ ការភ្ជាប់ Alcohol Sensor, Buzzer និង LED ជាមួយ board arduino Nano

១៥៥. Gas Methane Sensor

Gas Methane Sensor ប្រើសម្រាប់វាស់ ឬកំណត់កម្រិតនៃ Gas ដែលអាចនឹងឆ្លាយនៅក្នុងផ្ទះបាយ របស់យើង។ Gas Methane Sensor ឱ្យ OUTPUT Digital and Analog។



រូបទី១០.១២ ការភ្ជាប់ Gas Methane Sensor, Buzzer និង LED ជាមួយ board arduino Nano

ឧទាហរណ៍ ចូរសរសេរកូដបញ្ជាឱ្យអំពូល LED ពណ៌ក្រហមភ្លឺ និងមានសម្លេង Buzzer ឮពីដង្វាយ នៅពេលដែលមាន Gas។ បញ្ជាឱ្យអំពូល LED ពណ៌បៃតងភ្លឺ និងមិនមានសម្លេង Buzzer នៅពេលគ្មាន Gas ដោយកំណត់ជើងដូចក្នុងរូបទី១០.១២ និងមានការបង្ហាញទិន្នន័យនៅលើ Serial Monitor ផងដែរ។

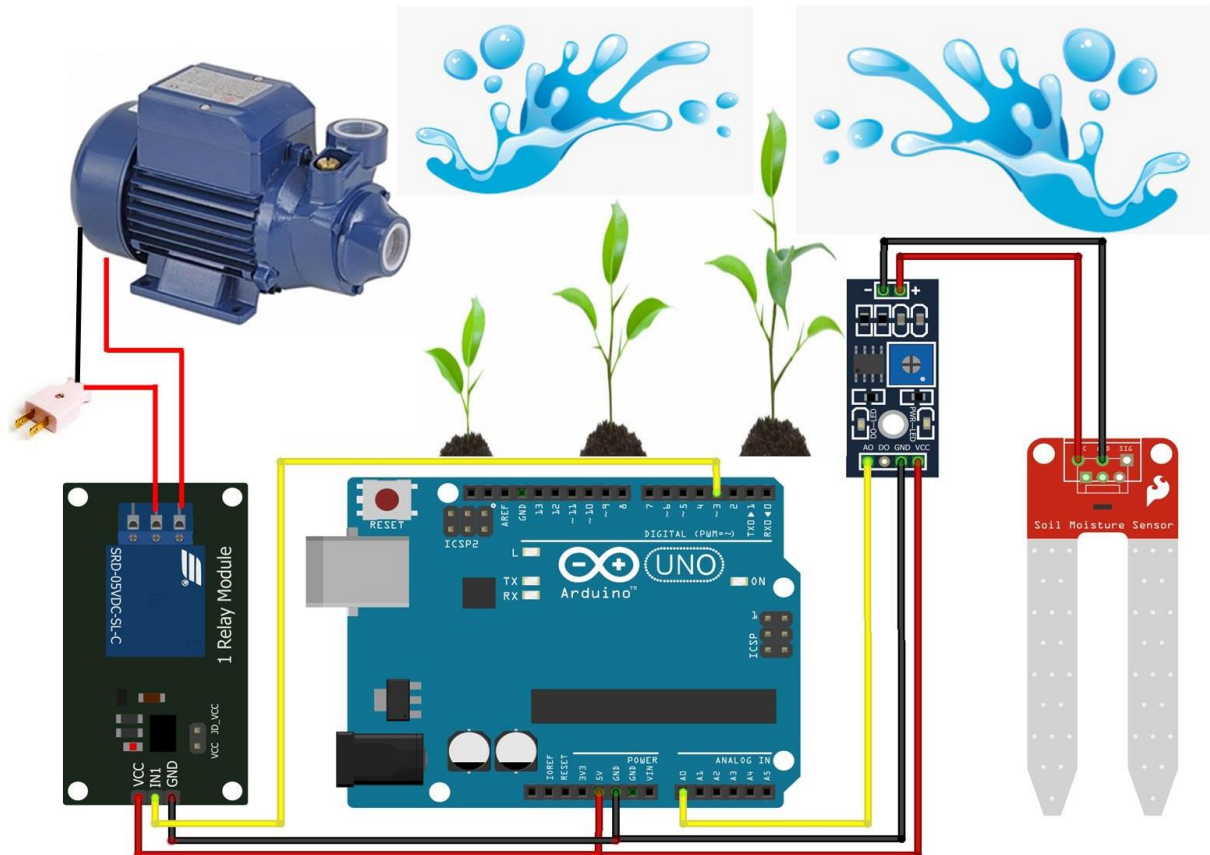
```
int redLed = 12;
int greenLed = 11;
int buzzer = 10;
int smokeA0 = A5;
int sensorThres = 400;
void setup( ) {
    pinMode( redLed, OUTPUT);
    pinMode( greenLed, OUTPUT );
    pinMode( buzzer, OUTPUT);
    pinMode( smokeA0, INPUT );
    Serial.begin( 9600 );
}
void loop( ) {
    int analogSensor = analogRead( smokeA0 );
    Serial.print( "Pin A0: " );
    Serial.println( analogSensor );
    if ( analogSensor > sensorThres )
    {
        digitalWrite( redLed, HIGH );
        digitalWrite( greenLed, LOW );
        tone( buzzer, 1000, 200 );
    }
    else {
        digitalWrite( redLed, LOW );
        digitalWrite( greenLed, HIGH );
        noTone( buzzer );
    }
    delay( 100 );
}
```

៤. Agriculture Sensor

ក. Soil Sensor

ជាប្រភេទ Sensor ដែលត្រូវបានគេប្រើសម្រាប់វាស់កម្រិតសំណើមនៅក្នុងដីបាន។ OUTPUT របស់ Soil Sensor ជា Digital and Analog ដែលតម្លៃស្មើ 1023 នៅពេលគ្មានសំណើម និងស្មើ 0 នៅពេលដីសើម 100%។

ឧទាហរណ៍ ចូរធ្វើប្រព័ន្ធស្រោចស្រពដំណាំ នៅពេលដែលដីស្ងួត ឱ្យស្រោចទឹក និងពេលដីមានសំណើមគ្រប់គ្រាន់ឱ្យឈប់ ស្រោចទឹក។



រូបទី១០.១៣ ការភ្ជាប់ Soil Sensor និង Relay ជាមួយ board arduino Uno

```
int RelayOnPump = 3;
int SoilSensor = A0;
int sensorValue = 0;

void setup( ) {
    Serial.begin( 9600 );
    pinMode( RelayOnPump, OUTPUT );
    pinMode ( SoilSensor, INPUT );
}

void loop( ) {
```

```

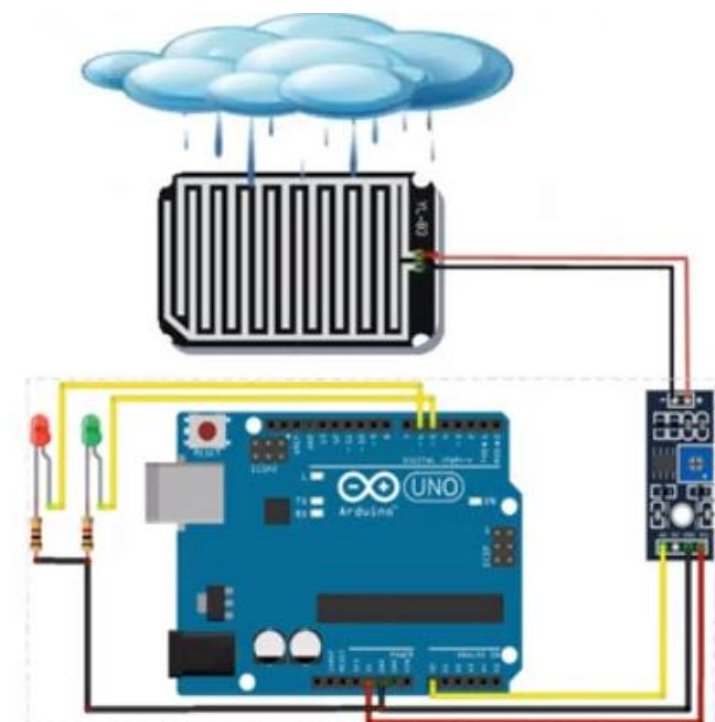
sensorValue = analogRead( A0 );
if ( sensorValue >= 600 && sensorValue <= 800 )
{
    Serial.println( sensorValue );
    digitalWrite( RelayOnPump, HIGH );
}
else {
    Serial.println( sensorValue );
    digitalWrite( RelayOnPump, LOW );
}
delay( 1000 );
}

```

ខ. Rain Sensor

ប្រើសម្រាប់ចាប់យកស៊ីញ៉ាល់ទៅពេលមានតំណក់ទឹកភ្លៀងស្រក់ចំ Sensor។ Rain Sensor ឱ្យ OUTPUT ជា Digital ។

ឧទាហរណ៍ ចូរសរសេរកូដបញ្ជាឱ្យអំពូល LED ពណ៌ក្រហមភ្លឺនឹងនៅលើ Serial Print ចេញពាក្យថា Rain នៅពេលមានទឹកភ្លៀង។ ឱ្យអំពូល LED ពណ៌បៃតងភ្លឺនឹងនៅលើ Serial Print ចេញពាក្យថា NO Rain នៅពេលគ្មានទឹកភ្លៀង។



រូបទី១០.១៤ ការតភ្ជាប់ Rain Sensor នឹង LEDx2 ជាមួយ board arduino Uno

```

int rainPin = A0;
int greenLED = 5;
int redLED = 6;
int SenosrholdValue = 500;
int SensorValue = 0;
void setup( ) {
    pinMode( rainPin,INPUT );
    pinMode( greenLED,OUTPUT );
    pinMode( redLED,OUTPUT );
    digitalWrite( greenLED, LOW );
    digitalWrite( redLED, LOW );
    Serial.begin( 9600 );
}
void loop( )
{
    SensorValue = analogRead( rainPin );
    Serial.print( SensorValue );
    if ( SensorValue < SenosrholdValue ) {
        Serial.println( "Rain " );
        digitalWrite( greenLED, LOW );
        digitalWrite( redLED, HIGH );
    }
    else
    {
        Serial.println( " NO Rain " );
        digitalWrite( greenLED, HIGH );
        digitalWrite( redLED, LOW );
    }
    delay( 500 );
}

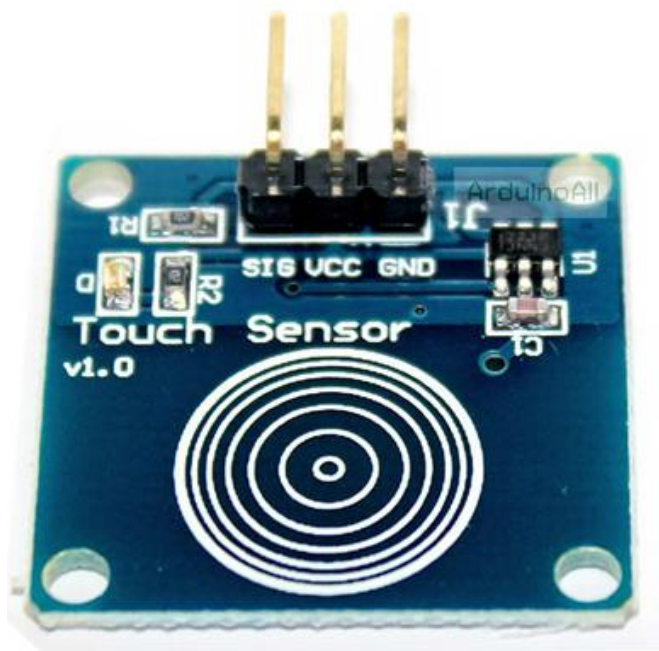
```

៥ . Switch sensor

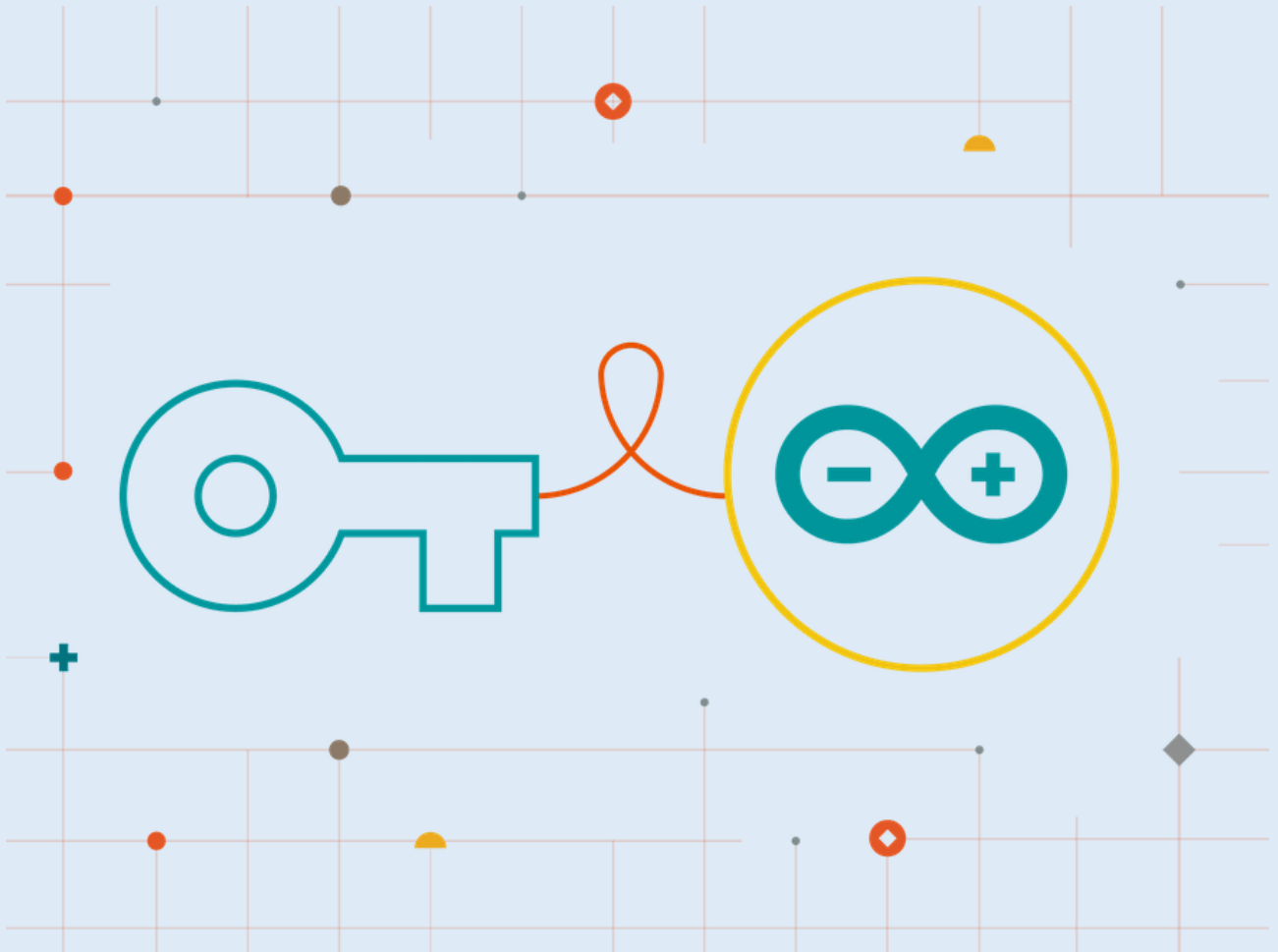
ក. Capacitive touch switch

ជាប្រភេទ កុងតាក់ដែលគ្មានមេកានិច តែវាប្រើ Capacitive ស្ពានដែលបស់យើងសម្រាប់ភ្ជាប់កុងតាក់។
ឧទាហរណ៍ ចូរសរសេរកូដបញ្ជាឱ្យ LED ភ្លឺនិងចេញពាក្យថា Touch នៅពេលមានការចុចលើ Sensor ហើយឱ្យ LED រលត់និងចេញពាក្យថា Un Touch នៅពេលគ្មានការចុចលើ Sensor ។

```
int LED = 2;
const int TouchPin = 9;
void setup( ) {
    pinMode(TouchPin, INPUT);
    pinMode(LED, OUTPUT);
    Serial.begin( 9600 );
}
void loop( ) {
    int sensorValue = digitalRead( TouchPin );
    if( sensorValue==1 )
    {
        Serial.println( " Un Touch " );
        digitalWrite( LED, LOW );
    }
    else
    {
        Serial.println( " Touch " );
        digitalWrite( LED, HIGH );
    }
    delay( 500 );
}
```



រូបទី១០.១៥ Capacitive touch switch



កុំភ្លេចលេងៗ ចង់ចេះមែនទែន