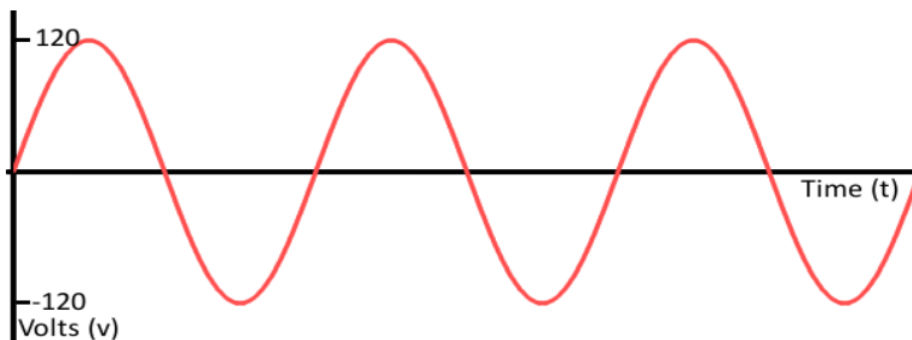


មេរៀនទី៣

Basic Digital & Basic Programming

I. អនាឡូក (ANALOG)

Analog ជាស៊ីញ៉ាល់ដែលគេប្រើប្រាស់សម្រាប់បញ្ចូលទិន្នន័យ ដែលនៅក្នុងការប្រើប្រាស់ជាមួយ Arduino Development Board តម្លៃរបស់ស៊ីញ៉ាល់ Analog មានចន្លោះពី 0 ដល់១០២៣ (10Bit)។

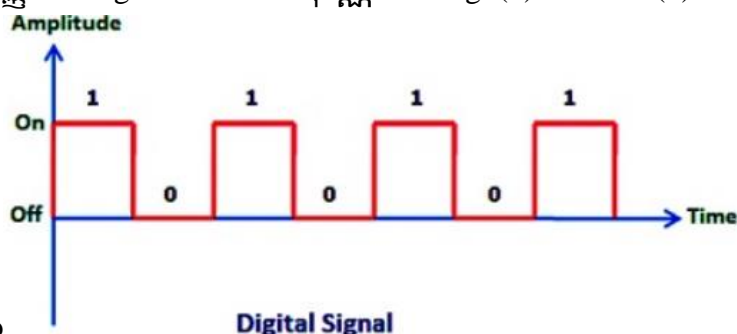


<https://learn.sparkfun.com/tutorials/analog-vs-digital/all>

នៅក្នុងប្រើប្រាស់ Arduino Development Board ជាមួយ Sensor ប្រភេទ Analog គេនិយមប្រើប្រាស់ជាការកំណត់តម្លៃនៅក្នុងចន្លោះណាមួយដើម្បីអោយមានប្រតិបត្តិការណាមួយកើតឡើង។ ឧទាហរណ៍ដូចជា LDR ដែលជាប្រភេទ Sensor មានតម្លៃប្រែប្រួលអាស្រ័យទៅលើកកម្រិតពន្លឺ គេតែងតែកំណត់តម្លៃ analog ដែលទទួលបានពីខាង LDR ដើម្បីទៅបញ្ជា (controller) ទៅលើ Output Device ណាមួយ ដូចជាពេលយប់ធ្វើអោយអំពូលភ្លឺជាដើម។

II. ឌីជីតល (DIGITAL)

digital ជាស៊ីញ៉ាល់ដែលគេប្រើប្រាស់សម្រាប់បញ្ចូលទិន្នន័យដូចគ្នាទៅនឹង analog ដែរ ប៉ុន្តែតម្លៃរបស់ស៊ីញ៉ាល់ Digital មានតែពីរប៉ុណ្ណោះគឺ High(1) និង Low(0)។



នៅក្នុងការប្រើប្រាស់ Arduino Development Board ជាមួយ Sensor ប្រភេទ Digital គេតែងតែប្រើប្រាស់ជាការកំណត់លក្ខខណ្ឌបែបពិត(Hight 1) ឬមិនពិត(Low:0)នៅពេលដែលលក្ខខណ្ឌពិត(Hight :1)តម្រូវអោយ Output Device ណាមួយដំណើរការធ្វើអ្វីមួយ។ ជាក់ស្តែងដូចជា Motion sensor ឬ PIR Sensor ជាប្រភេទ Digital Sensor ដែលចាប់សកម្មភាពមនុស្ស សត្វ ឬក៏អ្វីមួយដែលមានចលនា។ ឧទាហរណ៍ពេលមានមនុស្សដើរកាត់ Motion Sensor ផ្នែកខាង Controller នឹងទទួលបានទិន្នន័យពីខាង Sensor រួចបញ្ជាអោយអំពូលភ្លឺជាដើម។

III. VARIABLE TYPE

វាជាការកំណត់ប្រភេទរបស់អថេរ ដែលការកំណត់នេះវានឹងធ្វើអោយប្រភេទអថេរនោះមានតម្លៃជាក់លាក់ណាមួយ។ ការកំណត់អថេរនេះវាមានសារៈសំខាន់ណាស់ ដែលយើងត្រូវដឹងអោយច្បាស់ថា តើតម្លៃអថេរដែលយើងបានកំណត់នេះយើងយកទៅប្រើប្រាស់នៅក្នុងលក្ខខណ្ឌបែបណា បើសិនជាយើងកំណត់តម្លៃអថេរខុសពីលក្ខខណ្ឌដែលយើងនឹងយកមកប្រើប្រាស់ វានឹងធ្វើអោយប៉ះពាល់ដល់ Program ដែលយើងបានសរសេរ។

ប្រភេទតម្លៃលេខ	Bytes	តម្លៃកំណត់	ពន្យល់
Int	2	-32768 ទៅ 32767	ប្រើប្រាស់ជាមួយចំនួនគត់ ចំនួនវិជ្ជមាន និងអវិជ្ជមាន
Unsigned int	2	0 ទៅ 65535	ខុសគ្នាពី int ត្រង់ថាប្រើប្រាស់បានតែចំនួនវិជ្ជមាន
Long	4	-2147483648 ទៅ 2147483648	ដូចគ្នានឹង int តែខុសត្រង់ថា Long ចំនួនឡើងដល់ខ្ទង់ពាន់លាន
Unsigned long	4	0 ទៅ 4294967295	ខុសគ្នាពី Long ត្រង់ថាប្រើប្រាស់បានតែចំនួនវិជ្ជមាន
Float	4	-3.4028235E+38ទៅ 3.4028235E	ប្រើប្រាស់ជាមួយអថេរដែលតម្រូវតម្លៃលេខជាទសភាគ
Double	4		ដូចគ្នាទៅនឹង float ដែលខុសគ្នាត្រង់តម្លៃលេខរបស់ double មានកម្រិតលម្អៀងតិចតួច
Char	1	-128 ទៅ 127	ប្រើប្រាស់ជាមួយតួអក្សរមួយៗ
Boolean	1	1ពិត 0 មិនពិត	ប្រើប្រាស់ជាមួយតម្លៃមិនពិត

https://www.tutorialspoint.com/cplusplus/cpp_data_types.htm

ចំពោះអញ្ញាត្តិដែលមានតម្លៃថេរ ហើយស្ថិតនៅក្នុងប្រភេទអញ្ញាត្តិខាងលើត្រង់បានគេប្រើប្រាស់ Keyword មួយនៅពីមុខគឺ Const ។ ឧទាហរណ៍ Const int, Const float,.....

IV FURTHER SYNTAX

ប្រភេទ Further Syntax	មុខងារ
;(ចុចក្បៀស)	ប្រើប្រាស់នៅពេលដែលបញ្ចប់ Statement
{ } (ធ្មៀប)	សម្រាប់ប្រើប្រាស់នៅពេលដែលបើកអនុគមន៍ និង បិទអនុគមន៍
// (Comment មួយជួរ)	សម្រាប់បិទកូដក្នុងជួរណាមួយ ឬក៏ទុកសរសេរចំណាំនៅក្នុង Program
/**/ (comment)	ខុសគ្នាពី // ត្រង់ថាវាប្រើសម្រាប់បិទច្រើនជួរ
#define	ប្រើប្រាស់សម្រាប់ការប្រកាសនិមិត្តសញ្ញា
#include	ការទាញយក Library មកប្រើប្រាស់នៅក្នុង Sketch ដែលយើងកំពុងសរសេរ

V. ARITHMETIC OPERATOR

+,-,*,/,% ទាំងអស់នេះជា Arithmetic Operator ដែលប្រើប្រាស់សម្រាប់ធ្វើប្រមាណវិធី ឬក៏ ដក គុណ ចែកនិង ចែកយកសំណល់។

VI Comparison Operator

ប្រភេទ Comparison Operator	ការប្រើប្រាស់
==	ប្រើប្រាស់សម្រាប់ប្រៀបធៀបតម្លៃលេខពីរស្មើគ្នា
!=	ប្រើប្រាស់សម្រាប់ប្រៀបធៀបតម្លៃលេខផ្ទុយគ្នា
<	ប្រើប្រាស់សម្រាប់ប្រៀបធៀបតម្លៃតូចជាង
>	ប្រើប្រាស់សម្រាប់ប្រៀបធៀបតម្លៃធំជាង
<=	ប្រើប្រាស់សម្រាប់ប្រៀបធៀបតម្លៃតូចជាង ឬស្មើ
>=	ប្រើប្រាស់សម្រាប់ប្រៀបធៀបតម្លៃធំជាងឬស្មើ

VI BITWISE OPERATOR

អនុគមន៍តក្កវិទ្យា AND, OR, NOT, XOR, Shift Left, Shift Right ដែលទាំងអស់នេះវានឹងកើតមានឡើងនៅពេលប្រភេទតម្លៃជា bit គឺ 0 និង 1 ដែលគេហៅថា Logic Gate ។

-And Gate

A and B=A.B

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

-OR GATE

A OR B=A+B

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

NOT GATE

-NOT (A)= \hat{A}

A	\hat{A}
0	1
1	0

XOR GATE

A XOR B=A ⊕ B

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

Arduino Symbol	
AND	&
OR	
NOT	~
XOR	^
Shift Left	<<
Shift Right	>>

VII Boolean Operator

Arduino Symbol		ការប្រើប្រាស់
AND	&&	ធ្វើការប្រើប្រាស់សម្រាប់ប្រៀបធៀបលក្ខខណ្ឌពីរ ឬច្រើនជាងនឹង បើលក្ខខណ្ឌណាមួយមិនពិត នោះលក្ខខណ្ឌទាំងអស់គឺ មិនពិត
OR		ធ្វើការប្រើប្រាស់សម្រាប់ប្រៀបធៀបលក្ខខណ្ឌពីរ ឬច្រើនជាងនឹងបើលក្ខខណ្ឌ ទាំងអស់នោះមិនពិត នោះលក្ខខណ្ឌទាំងអស់គឺមិនពិត
NOT	!	លក្ខខណ្ឌនេះពិត លុះត្រាតែតម្លៃដែលនឹងប្រៀបធៀបផ្ទុយពីគ្នា

VIII. Boolean Operation

Arduino Symbol		ការប្រើប្រាស់
++	X++;	បូក x ដោយតម្លៃ ១ តែលទ្ធផលទទួលបានគឺ យកតម្លៃចាប់មុននឹងបូក
	++X;	បូក x ដោយតម្លៃ ១ តែលទ្ធផលទទួលបានគឺយកតម្លៃដែលបានបូកហើយ
--	X--;	ដក x ដោយតម្លៃ ១ តែលទ្ធផលទទួលបានគឺយកតម្លៃចាប់មុននឹងដក
	--X;	ដក x ដោយតម្លៃ ១ តែលទ្ធផលទទួលបានគឺយកតម្លៃដែលបានដកហើយ
+=	X+=Y;	X=X + Y;
-=	X-=Y;	X=X - Y;
=	X=Y;	X=X * Y;
/=	X/=Y;	X=X / Y;
%=	X%=Y;	X=X % Y;
&=	X&=Y;	X=X & Y;
=	X =Y;	X=X Y;



The End