Référence étendue du langage Arduino

Structure

Fonctions de base

Ces deux fonctions sont obligatoires dans tout programme en langage Arduino :

- void setup()
- void loop()

Structures de contrôle

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

Syntaxe de base

- ; (point virgule)
- {} (accolades)
- // (commentaire sur une ligne)
- /* */ (commentaire sur plusieurs lignes)
- #define
- #include

Opérateurs arithmétiques

- = (égalité)
- + (addition)
- - (soustraction)
- * (multiplication)
- / (division)
- % (modulo)

Opérateurs de comparaison

- == (égal à)
- != (différent de)
- < (inférieur à)
- > (supérieur à)
- <= (inférieur ou égal à)
- >= (supérieur ou égal à)

Variables et constantes

Les variables sont des expressions que vous pouvez utiliser dans les programmes pour stocker des valeurs, telles que la tension de sortie d'un capteur présente sur une broche analogique.

Constantes prédéfinies

Les constantes prédéfinies du langage Arduino sont des valeurs particulières ayant une signification spécifique.

- HIGH | LOW
- INPUT | OUTPUT
- true | false

A ajouter : constantes décimales prédéfinies

Expressions numériques

- Expressions numériques entières
- Expressions numériques à virgule

Types des données

Les variables peuvent être de type variés qui sont décrits ci-dessous.

Synthèse des types de données Arduino

- boolean
- char
- byte
- int
- unsigned int
- long
- unsigned long
- float (nombres à virgules)
- double (nombres à virgules)
- Les chaînes de caractères
- Les tableaux de variables
- le mot-clé void (fonctions)
- word
- PROGMEM

Fonctions

Entrées/Sorties Numériques

- pinMode(broche, mode)
- digitalWrite(broche, valeur)
- int digitalRead(broche)

Entrées analogiques

- int analogRead(broche)
- analogReference(type)

Sorties "analogiques" (génération d'impulsion)

- analogWrite(broche, valeur) - PWM

Entrées/Sorties Avancées

- tone()
- noTone()
- shiftOut(broche, BrocheHorloge, OrdreBit, valeur)
- unsigned long pulseIn(broche, valeur)

Temps

- unsigned long millis()
- unsigned long micros()
- delay(ms)
- delayMicroseconds(us)

Math

- min(x, y)
- max(x, y)
- abs(x)
- constrain(x, a, b)
- map(valeur, toLow, fromHigh, toLow, toHigh)
- pow(base, exposant)
- sq(x)
- sqrt(x)

Pour davantage de fonctions mathématiques, voir aussi la librairie math.h : log, log10, asin, atan, acos, etc...

Opérateurs booléens

- && (ET booléen)
- || (OU booléen)
- ! (NON booléen)

Pointeurs

- * pointeur
- & pointeur

Opérateurs bit à bit

- & (ET bit à bit)
- |(OU bit à bit)
- ^(OU EXCLUSIF bit à bit)
- ~(NON bit à bit)
- << (décalage à gauche)
- >> (décalage à droite)

Voir également :

- Manipulation des Ports

Opérateurs composés

- ++ (incrémentation)
- -- (décrémentation) (à revoir)
- += (addition composée)
- -= (soustraction composée)
- *= (multiplication composée)
- /= (division composée)
- &= (ET bit à bit composé)
- |= (OU bit à bit composé)

Conversion des types de données

- char()
- byte()
- int()
- long()
- float()
- word()

Portée des variables et qualificateurs

- Portée des variables
- static
- volatile
- const

Utilitaires

- sizeof() (opérateur sizeof)

Référence

- Code ASCII (à finir)

Trigonométrie

- sin(rad)
- cos(rad)
- tan(rad)

Nombres randomisés (hasard)

- randomSeed(seed)
- long random(max)
- long random(min, max)

Bits et Octets

- lowByte()
- highByte()
- bitRead()
- bitWrite()
- bitSet()
- bitClear()
- bit()

Interruptions Externes

- attachInterrupt(interruption, fonction, mode)
- detachInterrupt(interruption)

Interruptions

- interrupts()
- noInterrupts()

Voir également la librairie interrupt.h.

Communication

- Serial