MEMO LANGAGE ARDUINO – www.mon-club-elec.fr – version Février 2012

Structure

Fonctions de base

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

void setup()
void loop()

Syntaxe de base

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

Structures de contrôle

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

Instructions préprocesseur

#define (pas de ; en fin de ligne !)
#include (pas de ; en fin de ligne !)

Opérateurs arithmétiques

= (affectation) pas confondre avec ==
+ (addition)
- (soustraction)
* (multiplication)
/ (division)
% (modulo) (reste division)

Opérateurs de comparaison

== (égal à) ne pas confondre avec =
!= (différent de)
< (inférieur à)
> (supérieur à)
<= (inférieur ou égal à)
>= (supérieur ou égal à)

Opérateurs booléens

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

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)
Ports accessibles par DDRB, PORTB, PINB

Opérateurs composés

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

Pointeurs

* pointeur
& pointeur

Variables et constantes

Constantes prédéfinies

HIGH | LOW
INPUT | OUTPUT
true | false

Expressions numériques entières

décimal : 123
binaire : B10011011
hexadécimal :0xF0F0
octal : 0173

Expressions numériques décimales

10.0 vaut 10
2.34E5 vaut 2.34 x 10^5
67e-12 vaut 67 x 10^{-12}

Synthèse des types des données

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

void
Binaire :
boolean : true/false
Entiers signés :
int : -32 768 / +32 767
long : -2 147 483 648 / +2 147 483 647
Entiers non signés :
byte : 0 / +255
unsigned int : 0 / +65535
unsigned long : 0/ +4 294 967 295
word : 0 / +65535
Nombres à virgules
float : -3.4028235E+38 / +3.4028235E+38
double (ne pas utiliser – idem float)
Caractères
char (code ASCII) : -128 / + 127
voir également classe String
Les tableaux de variables : type[]
PROGMEM

Conversion des types de données

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

Portée des variables et qualificateurs

const, static, volatile

Utilitaires

sizeof() (opérateur sizeof)

Fonctions « Core » Arduino

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, fromL, fromH, toL, toH)
pow(base, exposant)
sq(x) (carré de x)
sqrt(x) (racine carrée de x)

Trigonométrie

sin(rad)
cos(rad)
tan(rad)
Pour davantage de fonctions mathématiques, voir aussi la librairie math.h : log, log10, asin, atan, acos, etc...

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.

Classe Serial

begin()
available()
read()
flush()
print()
println()
write()

Formats print() / println()

print("abcd") ; // affiche "abcd"
print(123) ; // affiche "123"
print(78, BYTE); // affiche "N"
print(78, BIN) ; // affiche "1001110"
print(78, OCT); // affiche "116"
print(78, DEC); // affiche "78"
print(78, HEX); // affiche "4E"
print(1.23456, 0); // affiche "1"
print(1.23456, 2); // affiche "1.23"
print(1.23456, 4); // affiche "1.2346"

Classe Stream

available()
read()
flush()
find()
findUntil()
peek()
readBytes()
readBytesUntil()
parseInt()
parsefloat()
setTimeout()

MEMO LANGAGE ARDUINO – www.mon-club-elec.fr – version Février 2012

Classe String

String()
charAt()
compareTo()
concat()
endsWith()
equals()
equalsIgnoreCase()
getBytes()
indexOf()
lastIndexOf()
length()
replace()
setCharAt()
startsWith()
substring()
toCharArray()
toLowerCase()
toUpperCase()
trim()

Librairie LiquidCrystal (LCD)

Initialisation

LiquidCrystal()
begin()

Gestion de l'écran

clear()
display()
noDisplay()

Positionner du curseur

home()
clear()
setCursor()

Modifier l'aspect du curseur

cursor()
noCursor()
blink()
noBlink()

Ecriture

print()
write()

Contrôle du mode d'affichage

scrollDisplayLeft()
scrollDisplayRight()
autoscroll()
noAutoscroll()
leftToRight()
rightToLeft()?

Création de caractère personnalisé

createChar()

Servo

Servo
attach()
write()
writeMicroseconds()
read()
boolean attached()
detach()

Stepper

Stepper(nombre_pas, broche1, broche2)
Stepper(nombre_pas, broche1, broche2, broche3, broche4)
setSpeed(vitesse)
step(nombre_pas)

SPI

begin()
end()
setBitOrder()
setClockDivider()
setDataMode()
transfer()

Wire / I2C

begin()
begin(adresse)
requestFrom(adresse, quantite)
beginTransmission(adresse)
endTransmission()
send()
available()
receive()
onReceive(fonction)
onRequest(fonction)

Ethernet

Classe Ethernet

La classe Ethernet initialise la librairie ethernet et les paramètres du réseau.

begin()
localIP()

Classe IP

IPAddress()

Classe Server

La classe Server crée des serveurs qui envoient des données vers et reçoivent des données depuis des clients connectés (programmes s'exécutant sur d'autres ordinateurs ou composants).

Server() / EthernetServer()
begin()
available()
write()
print()
println()

Classe Client

La classe Client crée des clients qui peuvent se connecter à des serveurs et envoyer ou recevoir des données.

Client() / EthernetClient()
connected()
connect()
write()
print()
println()
available()
read()
flush()
stop()

Software Serial

SoftwareSerial()
begin()
read()
print()
println()

*Keypad**

Keypad(makeKeymap(userKeymap), row[], col[], rows, cols)
begin()
getKey()
getState()
setHoldTime(unsigned int time)
setDebounceTime(unsigned int time)
addEventListener(keypadEvent)

Eeprom

Read()
write()

SD Card

Classe SD

La classe SD fournit les fonctions pour l'accès à la carte mémoire SD et la manipulation de ses fichiers et répertoires.

begin()
exists()
mkdir()
open()
remove()
rmdir()

Classe File

La classe File permet la lecture et l'écriture dans et depuis les fichiers individuels sur la carte mémoire SD.

available()
close()
flush()
peek()
position()
print()
println()
seek()
size()
read()
write()
isDirectory()
openNextFile()
rewindDirectory()

*One-Wire**

Initialisation

OneWire myWire(pin)
myWire.search(addrArray)
myWire.reset_search()
myWire.skip()

Communication

myWire.reset()
myWire.select(addrArray)
myWire.write(num)
myWire.write(num, 1)
myWire.read()

Contrôle des données

myWire.crc8(dataArray, length)

Contrôle de de l'alimentation

myWire.depower()?

Autres

Firmata...

* Librairie fournies par la communauté ne faisant par partie des librairies officielles Arduino.

Comptage binaire/hexa

DEC	BIN	HEX	DEC	BIN	HEX
0	B0000	0x0	8	B1000	0x8
1	B0001	0x1	9	B1001	0x9
2	B0010	0x2	10	B1010	0xA
3	B0011	0x3	11	B1011	0xB
4	B0100	0x4	12	B1100	0xC
5	B0101	0x5	13	B1101	0xD
6	B0110	0x6	14	B1110	0xE
7	B0111	0x7	15	B1111	0xF