ACTIVITÉ - Communication par BUS I²C

P²C BUS

ATRIUM PLANTE TOUJOURS, JE DÉCIDE DE JOINDRE MES ÉLÈVES SUR FORTNITE...



ETAPE 1 : PROGRAMMATION DE LA COMMUNICATION I2C AVEC ARDUINO

La bibliothèque **Wire** permet de communiquer sur le bus l²C en Arduino. Il est nécessaire de l'importer dans votre programme :

#include <Wire.h>

Description des fonctions I2C :

Wire.begin() : initialise le mode de communication I2C. À écrire une fois dans la procédure setup()

Wire.beginTransmission(adress) : débute une transmission avec un circuit I2C. Cette fonction permet de prendre la main sur le bus, et d'indiquer, par la variable adress, avec quel circuit l'on désire communiquer.

Wire.requestFrom(adress, quantity, STOP):

Requête envoyée à l'esclave (*adress*) lui demandant de placer sur le bus I2C les données de son ou de ses registres. Le bit R/W est alors automatiquement mis à 1 (Lecture).



Quantity correspond au nombre **d'octets** à lire. **STOP** est un booléen (True/False) indiquant si l'on doit placer un stop après exécution de la requête pour libérer le bus.

- Wire.available() : vérifie si les données sont disponibles sur le BUS après le lancement d'une requête. Cette fonction renvoie le nombre d'octets disponibles.
- Wire.read(): Permet de lire un ou plusieurs octets placés par l'esclave sur le bus. Cette fonction ne peut fonctionner que si un début de transmission est effectué et si les données sont disponibles.
- Wire.endTransmission(): Termine une transmission en cours sur le bus I2C. Cette fonction permet de libérer le bus en plaçant un STOP sur la trame.

Wire.write(byte) : Permet d'écrire un octet byte sur le bus. Cette fonction ne peut fonctionner que si un début de transmission est effectué. Le bit R/W est alors automatiquement mis à 0.

Exemple de Programme en Écriture :



Ce qui se traduit en Arduino par :



Exemple de programme en Lecture :



Ce qui se traduit en Arduino par :

Wire.beginTransmission(DEV_ID); // START - Prise en main du BUS et connexion au circuit DS1621 Wire.write(0xAA); // Commande de lecture de température Wire.requestFrom(DEV_ID, 2, true); //Lecture de 2 octets if (Wire.available()) { //Si les données sont disponibles... firstByte = Wire.read(); //lecture lere Octet secondByte = Wire.read(); //lecture 2eme Octet (1/2 degré) } delay(10); Wire.endTransmission(); //STOP - Libération du BUS



♦ RÉALISATION du PROGRAMME :

Ajouter des commentaires sur le programme maitre Arduino.

#include <Wire.h>

//.... const int SLAVE_ADDR1 = 5; //... const int PIN_BUTTON1 = 2;

//....const int PIN_LED1 = 10;

//button pressed command (message) to send to slave const char BTN_PRESSED = 'H'; //button released command (message) to send to slave const char BTN_RELEASED = 'L';

//button state variable
int button1State = 0;

bool pressed1 = false;

void setup(){
//

Wire.begin();

//set button pins to input mode - we are reading button state
pinMode(PIN_BUTTON1,INPUT_PULLUP);

//set master led pins to output - we are controling leds
pinMode(PIN_LED1, OUTPUT);

}

ł

}

void loop()

//on each loop of master controller:

```
//read button states
button 1State = digitalRead(PIN_BUTTON1);
```

if ((button1State == LOW) && (!pressed1))

//turn on master (local) led digitalWrite(PIN_LED1, HIGH); //send slave message - button is pressed Wire.beginTransmission(SLAVE_ADDR1); Wire.write(BTN_PRESSED); Wire.endTransmission(); pressed1 = true;



TP Bus I2C

else if ((button1State == HIGH) && pressed1)	
{	
//Turn off own led	
digitalWrite(PIN_LED1, LOW);	
//send slave message - button is released	
Wire.beginTransmission(SLAVE_ADDR1);	
Wire.write(BTN_RELEASED);	
Wire.endTransmission();	
pressed1 = false;	
}	
delay(50);	
}	

- **4** Donner l'adresse de l'arduino esclave ?
- ↓ Quelle sont les 2 valeurs envoyé sur le bus I2C ?
- ↓ Ouvrir le programme BUSIC.EXE et simuler la ligne de programme suivante :

Wire.beginTransmission(SLAVE_ADDR1); // adresse sur 7 bits

Faire une impression écran du chronogramme.

↓ Ouvrir le programme BUSIC.EXE et simuler la ligne de programme suivante :

Wire.write(BTN_PRESSED); (lorsque le bouton est appuyé)

р						°°°	°°,	° , ' o	° , ,	¹ ° ₀	'°,	' '0	۱ ۱
	b4 1	b 3 1	Þ 2 †	Ь +	Row	0	I	2	3	4	5	6	7
	0	0	0	0	0	NUL .	DLE	SP	0	0	Р	`	р
	0	0	0	1	1	SOH	DC1	!	1	A	Q ·	0	P
	0	0	1	0	2	STX	DC2		2	B	R	b	r
	0	0	1	I	3	ETX	DC 3	#	3	C	S	С	5
	0	1	0	0	4	EOT	DC4	\$	4	D	т	d	t
	0	1	0	1	5	ENQ	NAK	%	5	E	U	e	U
	0	1	1	0	6	ACK	SYN	8	6	F	V	f	V
	0	-	1	1	7	BEL	ETB	•	7	G	₩	g	w
	1	0	0	0	8	BS	CAN	(8	н	x	h	×
	1	0	0	1	9	нт	EM)	9	1	Y	i	У
	1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
	1	0	I	1		VT	ESC	+	;	к	C	k	{
	1	1	0	0	12	FF	FS	•	<	L	\	1	1
	1	1	0	1	13	CR	GS	-	¥	м	J	m	}
	1	1	1	0	4	SO	RS		>	N	^	n	\sim
	1	1	1	1	15	S 1	US	1	?	0		0	DEL

USASCII code chart

Faire une impression écran du chronogramme.



```
#include <Wire.h>
//led pin number
int LED = 7;
void setup() {
 //....
 Wire.begin(5);
 //....
 Wire.onReceive(receiveEvent);
 //set let pin to output
 pinMode(LED, OUTPUT);
 //turn off slave led
 digitalWrite(LED, LOW);
 Serial.begin(9600);
}
void loop() {
//no work here, we just respond to received events. see function below.
}
//this function is called when I2C event is received
void receiveEvent(int howMany)
{
//....
char c = Wire.read();
//....
Serial.println(c);
if( c == 'H' ){
//turn slave led on
digitalWrite(LED,HIGH);
} else if ( c == 'L' ){
//turn slave led off
digitalWrite(LED,LOW);
}
}
```



• RÉALISATION du Schéma:

Réaliser le schéma suivant sur : tinkercad.com



Quelles sont les broches utilisées pour le bus I2C?

Simuler le fonctionnement du circuit, ouvrir le moniteur série de l'arduino esclave.



Améliorer le circuit

On veut rajouter une arduino esclave, un bouton 2 et une led verte.

- 4 Si on appuie sur le bouton 1 arduino esclave 1 allume led rouge.
- 4 Si on appuie sur le bouton 2 arduino esclave 2 allume led verte.



