

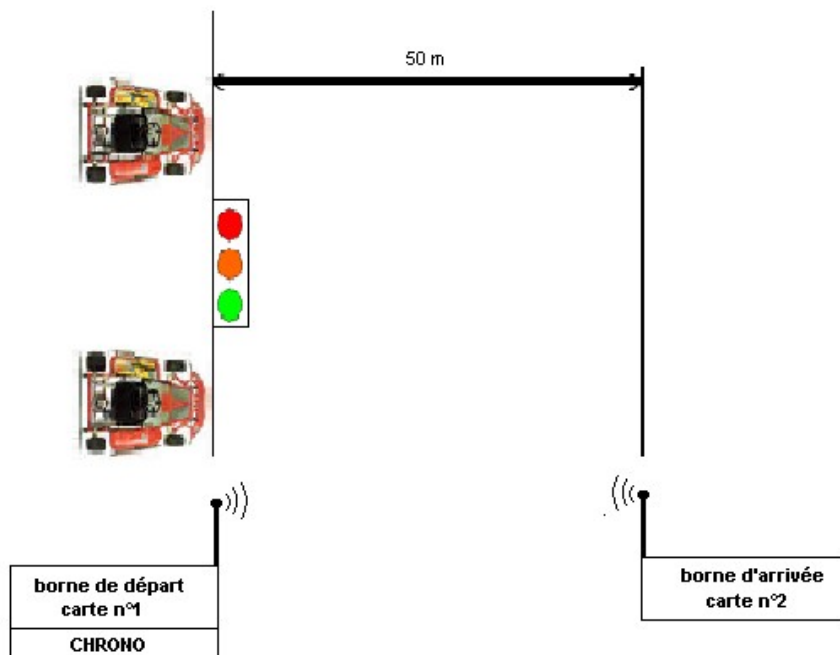
Communications sans fil

Balise de mesure de temps pour l'épreuve de 50 mètres départ arrêté : Carte micro-contrôleur

Le but de notre projet sera d'effectuer une communication entre deux cartes électroniques, en utilisant la technologie FM ou Wifi et la programmation d'un Atmega8535.

Ce projet sera adapté à l'épreuve du 50 mètres départ arrêté. Il s'agit d'une compétition entre deux karts.

Lorsqu'un kart coupe le faisceau laser de la borne de départ, la carte micro-contrôleur n°1 lance le chronomètre. Le kart effectue sa course jusqu'à ce qu'il coupe le faisceau laser de la borne d'arrivée. La carte micro-contrôleur n°2 envoie donc un signal à la carte n°1 pour lui dire de stopper le chronomètre. Le temps mis par les deux karts sera ensuite affiché sur un écran LCD.



Cahier des charges

Le cahier des nous imposes d'utiliser les cartes micro-contrôleur, comportant l'ATmega8535 et un afficheur LCD 4*16(lignes*caractères).

Le temps de communication entre les deux cartes doit être inférieur à 10ms (9600bauds). Nous disposons de 2 antennes FM, 2 modules d'émission et de 2 modules de réception ainsi que 2 cartes Wifi et de 2 modules WLAN selon la solution technologique retenue.

Type d'affichage souhaité sur l'afficheur LCD :

<i>ligne1</i>	<i>Temps au 50 m :</i>
<i>ligne2</i>	<i>Kart 1 : xx'xx''</i>
<i>ligne3</i>	<i>Kart 2 : xx'xx''</i>
<i>ligne14</i>	<i>Record : xx'xx''</i>

Il s'agira alors, après avoir succinctement étudié les différents modules, de programmer l'ATMega8535 en langage C dans le but de réaliser :

- Le protocole de communication.
- Les calculs de temps et l'interprétation des données reçues.
- L'affichage sur le LCD.

<i>Semaines Tâches</i>	3	4	5	6	7	8	9	10	11	12	13	14	15	16
<i>Découverte du sujet</i>														
<i>Recherches documentaires</i>														
<i>Mise en place de l'ordinogramme</i>														
<i>Codage et rectifications</i>														
<i>Test de l'ATmega8535</i>														
<i>Finalisation du projet (module Fm, module Wifi)</i>														
<i>Rédaction du dossier</i>														

Planning Prévisionnel

MONSTERLET Romain

BOUQUET Romain

Groupe P2

« Communication sans fil »

Test n2 : communication entre deux cartes (avec les fils)

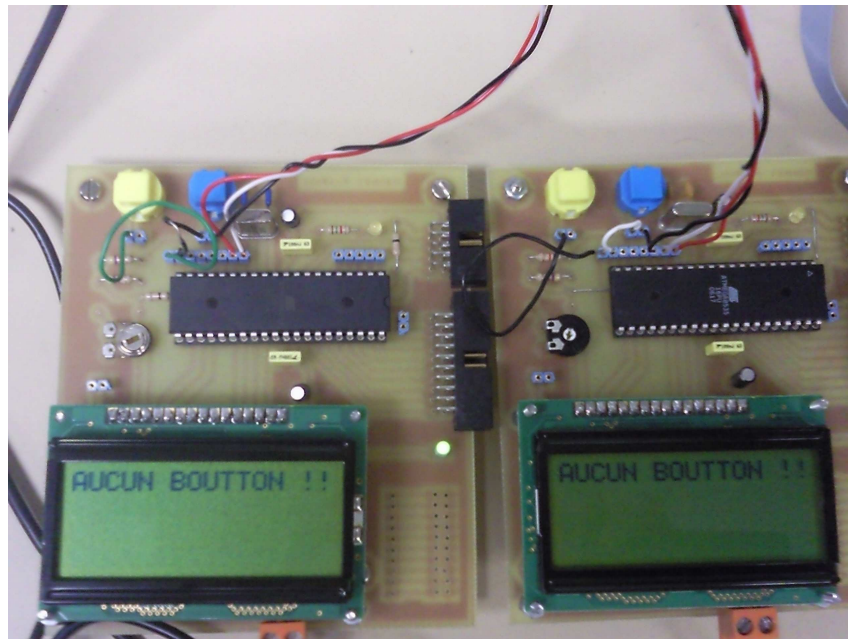


Illustration 1: Aucun bouton poussoir pressé

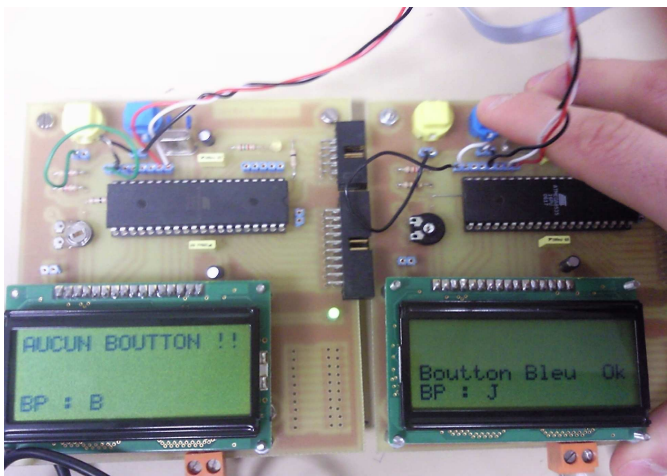


Illustration 2: Bouton Bleu pressé

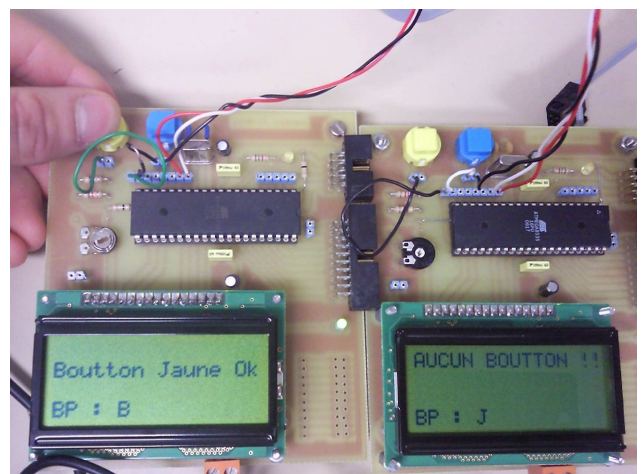


Illustration 3: Bouton Jaune pressé

CodeWizardAVR - untitled.cwp

File Help

USART | Analog Comparator | ADC | SPI
 I2C | 1 Wire | 2 Wire (I2C)
 LCD | Bit-Banged | Project Information
 Chip | Ports | External IRQ | Timers

Chip: ATmega8535
 Clock: 16,000,000 MHz

Check Reset Source
 Program Type: Application

CodeWizardAVR - untitled.cwp

File Help

I2C | 1 Wire | 2 Wire (I2C)
 LCD | Bit-Banged | Project Information
 Chip | Ports | External IRQ | Timers
 USART | Analog Comparator | ADC | SPI

Receiver Rx Interrupt
 Transmitter Tx Interrupt

Baud rate: 9600 x2
 Baud Rate Error: 0.2%
 Communication Parameters:
 8 Data, 1 Stop, No Parity
 Mode: Asynchronous

CodeWizardAVR - untitled.cwp

File Help

I2C | 1 Wire | 2 Wire (I2C)
 LCD | Bit-Banged | Project Information
 USART | Analog Comparator | ADC | SPI
 Chip | Ports | External IRQ | Timers

Port A | Port B | Port C | Port D

Data Direction	Pullup/Output Value
Bit 0	In T Bit 0
Bit 1	In T Bit 1
Bit 2	In T Bit 2
Bit 3	In T Bit 3
Bit 4	In T Bit 4
Bit 5	In T Bit 5
Bit 6	In T Bit 6
Bit 7	In T Bit 7

CodeWizardAVR - untitled.cwp

File Help

USART | Analog Comparator | ADC | SPI
 I2C | 1 Wire | 2 Wire (I2C)
 Chip | Ports | External IRQ | Timers
 LCD | Bit-Banged | Project Information

LCD Port: PORTC
 Chars./Line: 16

PORT Bit 0 - RS (LCD Pin 4)
 PORT Bit 1 - RD (LCD Pin 5)
 PORT Bit 2 - EN (LCD Pin 6)
 PORT Bit 3 - Free
 PORT Bit 4 - DB4 (LCD Pin 11)
 PORT Bit 5 - DB5 (LCD Pin 12)
 PORT Bit 6 - DB6 (LCD Pin 13)
 PORT Bit 7 - DB7 (LCD Pin 14)

Programme des deux Atmega8535

```
// Bouton poussoir ( PIND.6 et PIND.7 branché sur le Port D, pin 6 et 7 )
# define BP_BLEU PIND.7
#define BP_JAUNE PIND.6

//Code hexadécimal pour les lettres « o »; « B » « J »
# define o 0x30
#define J 0x4A
#define B 0x42

while (1)
{
    // Place your code here

    unsigned char i;
    unsigned char tampon[82];

    USART_Transmit(O);
    i=USART_Receive();

    if((BP_BLEU==1)&&(BP_JAUNE==0))
    {
        lcd_gotoxy(0,0);
        lcd_putsf("          ");
        lcd_gotoxy(0,1);
        lcd_putsf("Boutton Jaune Ok");
        lcd_gotoxy(0,2);
        lcd_putsf("          ");
        USART_Transmit(J);
    }
}
```

```

if((BP_BLEU==0)&&(BP_JAUNE==1))
{
    lcd_gotoxy(0,0);
    lcd_putsf("      ");
        lcd_gotoxy(0,1);
    lcd_putsf("      ");
        lcd_gotoxy(0,2);
    lcd_putsf("Boutton Bleu Ok");
    USART_Transmit(B);

}

```

```

if((BP_BLEU==1)&&(BP_JAUNE==1))
{

        lcd_gotoxy(0,0);
        lcd_putsf("AUCUN BOUTTON !!");
        lcd_gotoxy(0,1);
    lcd_putsf("      ");
    lcd_gotoxy(0,2);
    lcd_putsf("      ");

}

```

```

if(i!=0)
{
        sprintf(tampon,"BP : %c",i);
        lcd_gotoxy(0,3);
        lcd_puts(tampon);

}

```

```

};
}

```

**Programme n°1 complet (programmation d'un ATmega8535, gérer
l'affichage de l'écran LCD)**

/******

This program was produced by the
CodeWizardAVR V1.24.2c Professional
Automatic Program Generator
© Copyright 1998-2004 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.ro>
e-mail:office@hpinfotech.ro

Project :
Version :
Date : 26/01/2010
Author : F4CG
Company : F4CG
Comments:

Chip type : ATmega8535
Program type : Application
Clock frequency : 16,000000 MHz
Memory model : Small
External SRAM size : 0
Data Stack size : 128

*****/

```
#include <mega8535.h>
```

```
// Alphanumeric LCD Module functions
```

```
#asm
```

```
.equ __lcd_port=0x15 ;PORTC
```

```
#endasm
```

```
#include <lcd.h>
```

```
#include <delay.h>
```



```

// Declare your global variables here

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization (ENTREES ANALOGIQUES)
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization (PROGRAMMATION)
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization (LCD)
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTC=0x00;
DDRC=0xFF;

// Port D initialization (COMMUNICATION & BPs)
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=Out Func0=Out
// State7=P State6=P State5=T State4=T State3=T State2=T State1=0 State0=0
PORTD=0xC0;
DDRD=0x03;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected

```

```
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
```

```

MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x67;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
// Analog Comparator Output: Off
ACSR=0x80;
SFIOR=0x00;

// LCD module initialization
lcd_init(16);

while (1)
{
    // Place your code here

    if((PIND.6==1)&&(PIND.7==0))
    {

```

```
    lcd_gotoxy(0,0);
    lcd_putsf("Boutton Jaune Ok");
    delay_ms(2000);
    lcd_clear();

}
if((PIND.6==0)&&(PIND.7==1))
{
    lcd_gotoxy(0,0);
    lcd_putsf("Boutton Bleu Ok");
    delay_ms(2000);
    lcd_clear();

}
else{
    lcd_gotoxy(0,0);
    lcd_putsf("AUCUN BOUTTON !!");
}

};
}
```