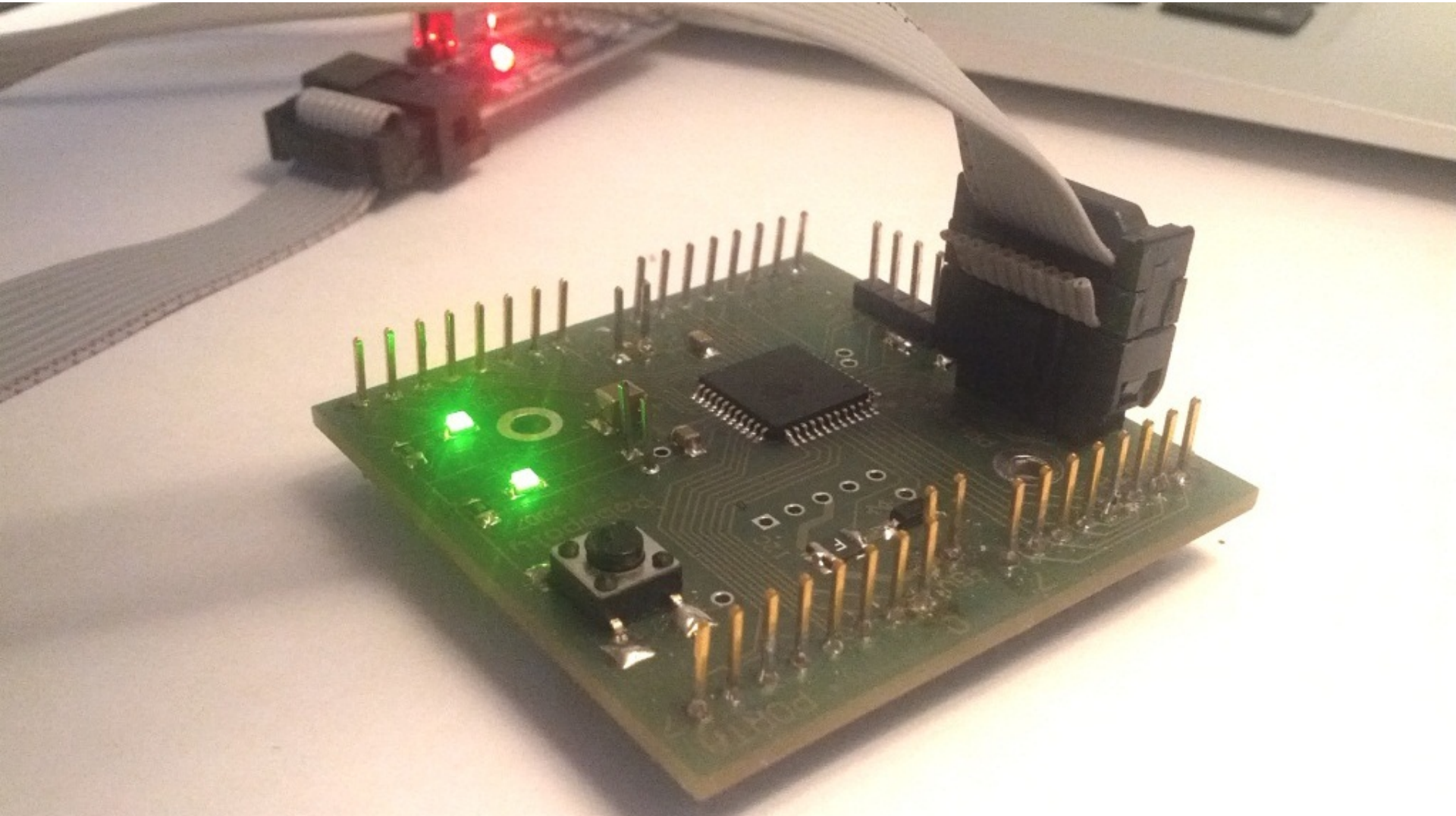


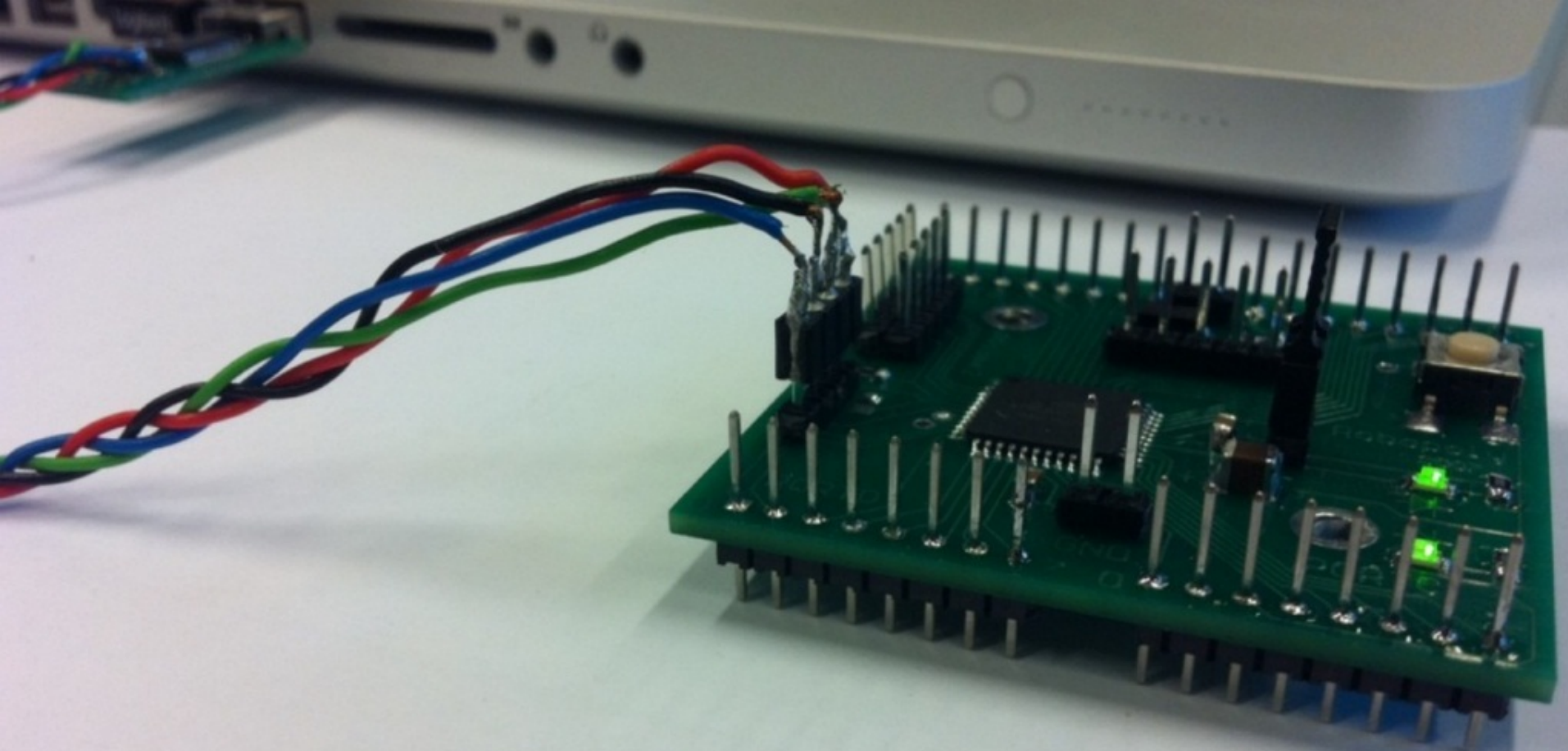


PROGRAMMATION

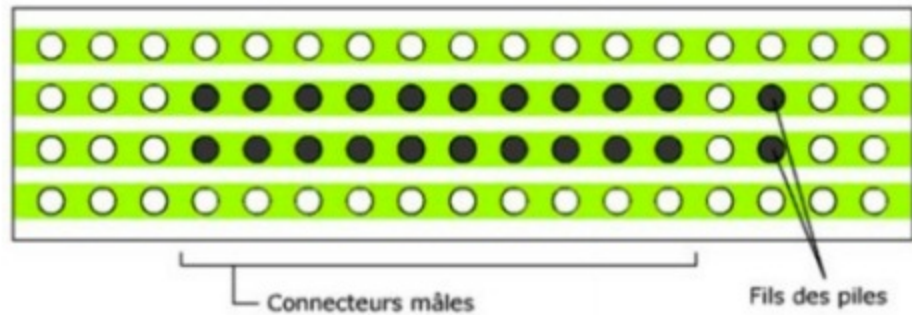
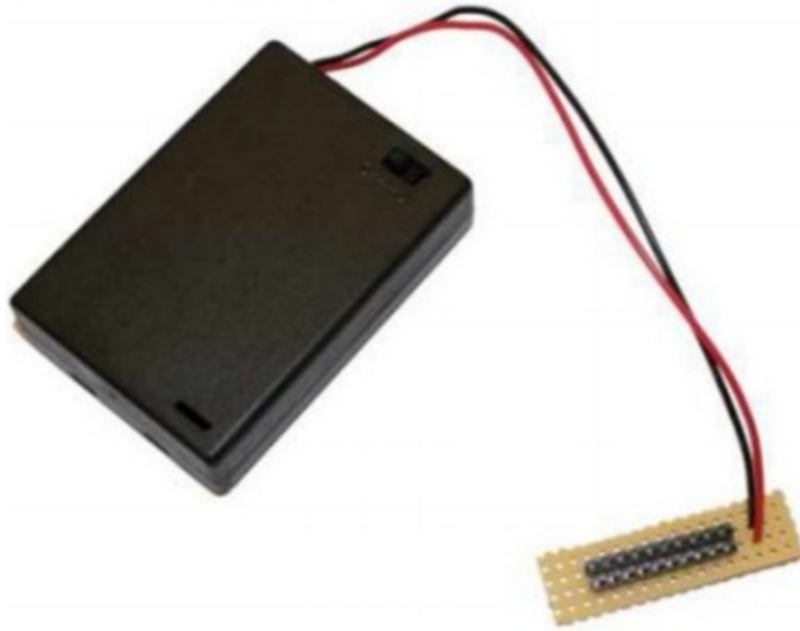
Bootloader



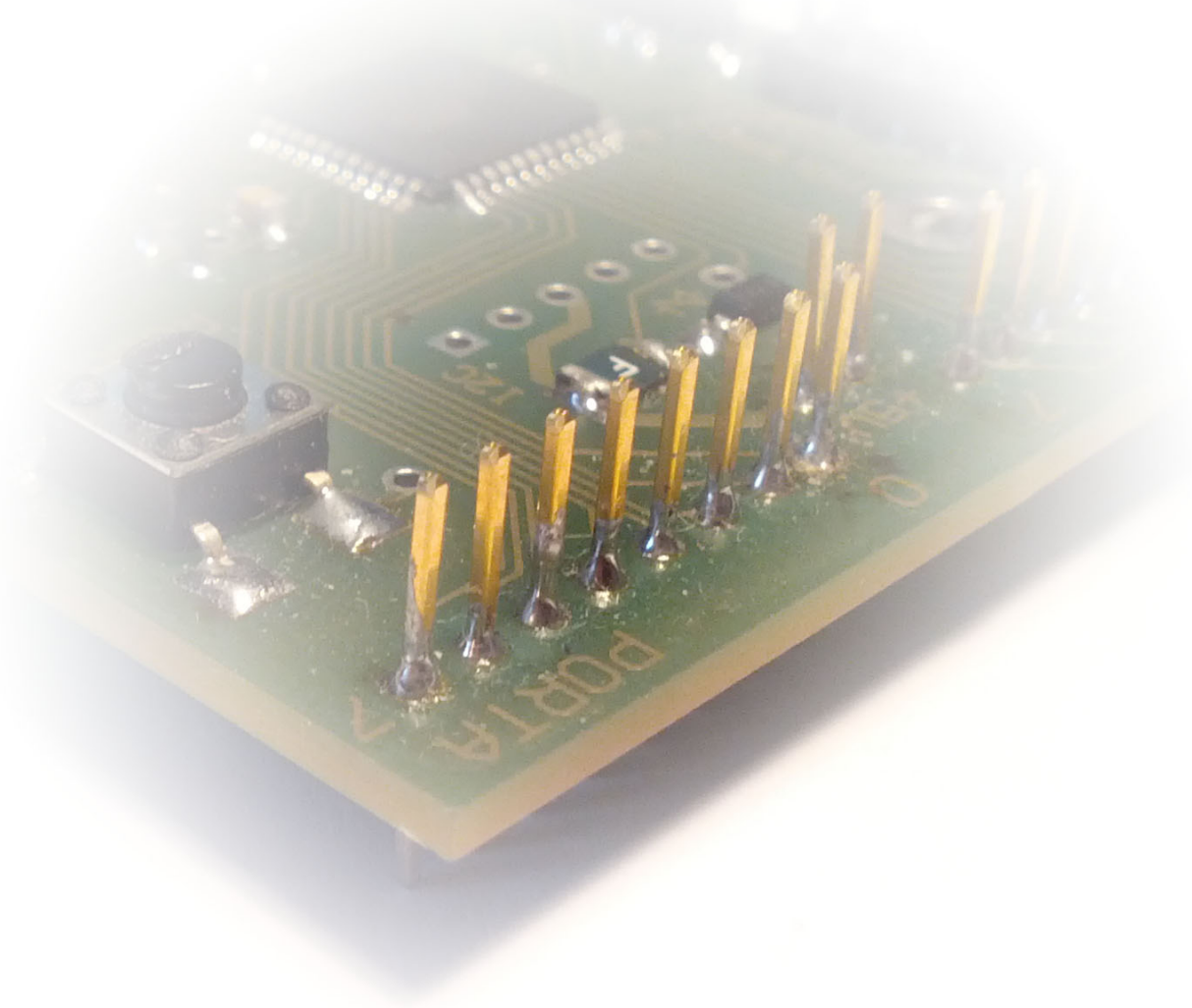
Port UART



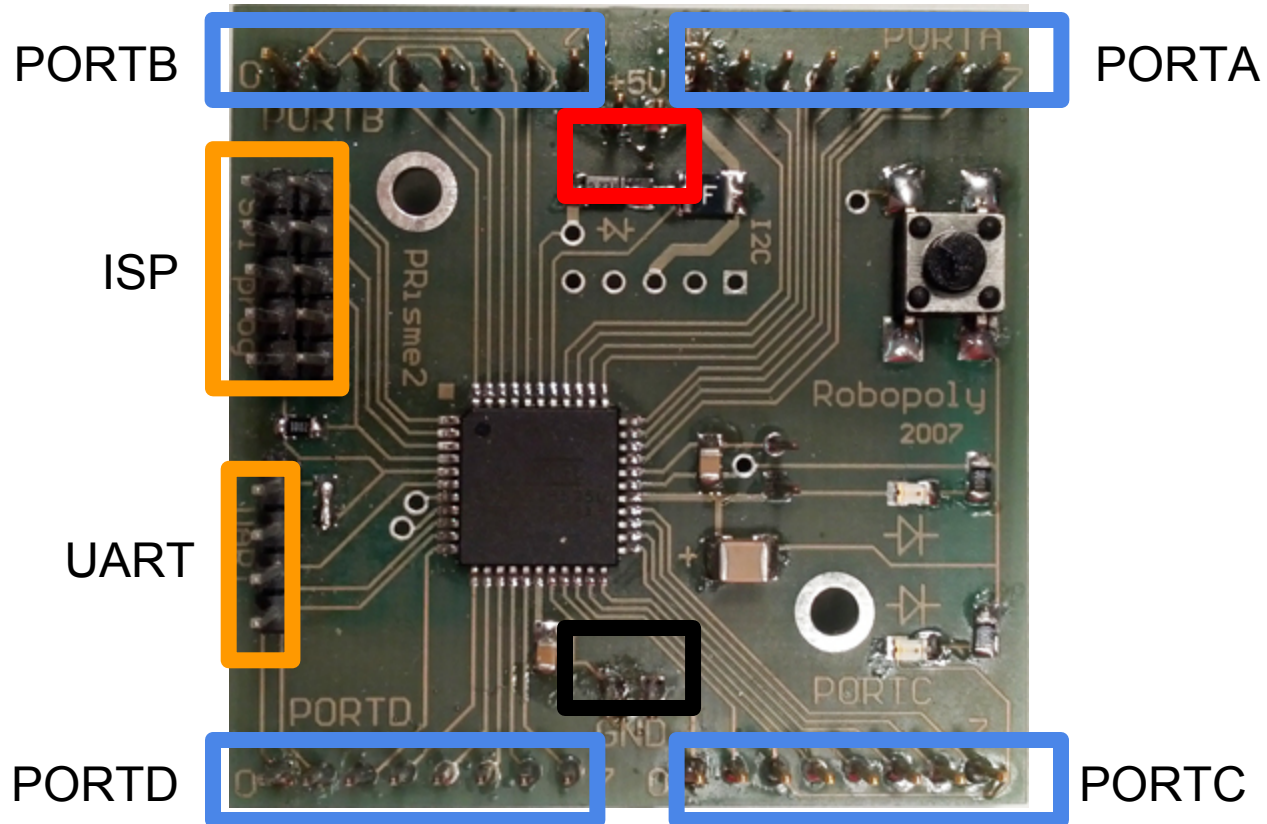
Alimentation



Un pin



Ports



Signal sur un pin



1

Connecté au même
potentiel que l'alimentation:
3.6V des batteries

0

Connecté à la
masse (0V)

?

Pas
connecté

Expert Mode



Arduino NG or older w/ ATmega8

PRisme2 8MHz at 9600 baud

✓ PRisme2 8MHz at 9600 baud (Expert Mode)

PRisme2 8MHz at 115200 baud

Programmation Arduino



```
#define LED PC(2)

void setup()
{
  // set pin as output
  pinMode(LED, OUTPUT);
}

void loop()
{
  // turn LED on
  digitalWrite(LED, HIGH);
  // in milliseconds
  delay(500);
  // turn LED off
  digitalWrite(LED, LOW);
  delay(500);
}
```

Programmation Arduino



constante

```
#define LED PC(2)
```

définir la pin du
LED en sortie

```
void setup()  
{  
  // set pin as output  
  pinMode(LED, OUTPUT);  
}
```

appelé une fois
(initialisation)

allumer la LED

```
void loop()  
{  
  // turn LED on  
  digitalWrite(LED, HIGH);  
  // in milliseconds  
  delay(500);  
  // turn LED off  
  digitalWrite(LED, LOW);  
  delay(500);  
}
```

appelé en boucle

attendre 500ms

éteindre la LED

Programmation Arduino



```
#define LED PC(2)

void setup()
{
  // set pin as output
  pinMode(LED, OUTPUT);
}

void loop()
{
  // turn LED on
  digitalWrite(LED, HIGH);
  // in milliseconds
  delay(500);
  // turn LED off
  digitalWrite(LED, LOW);
  delay(500);
}
```

946 octets

Programmation en C



```
#include <avr/io.h>
#include <util/delay.h>

int main()
{
    DDRC = 0b00000100;
    while(1)
    {
        PORTC = 0b00000000;
        _delay_ms(500);
        PORTC = 0b00000100;
        _delay_ms(500);
    }

    return 0;
}
```



Programmation en C

En-têtes

```
#include <avr/io.h>
#include <util/delay.h>
```

définir la pin du
LED en sortie

```
int main()
{
  DDRC = 0b00000100;
  while(1)
```

éteindre la LED

```
{
  PORTC = 0b00000000;
  _delay_ms(500);
```

allumer la LED

```
  PORTC = 0b00000100;
  _delay_ms(500);
}
```

```
  return 0;
}
```

appelé en boucle

attendre
500ms

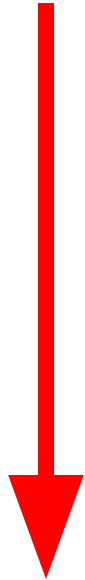
Programmation en C



```
#include <avr/io.h>
#include <util/delay.h>

int main()
{
    DDRC = 0b00000100;
    while(1)
    {
        PORTC = 0b00000000;
        _delay_ms(500);
        PORTC = 0b00000100;
        _delay_ms(500);
    }

    return 0;
}
```



Programmation en C



```
#include <avr/io.h>
#include <util/delay.h>

int main()
{
    DDRC = 0b00000100;
    while(1)
    {
        PORTC = 0b00000000;
        _delay_ms(500);
        PORTC = 0b00000100;
        _delay_ms(500);
    }

    return 0;
}
```

146 octets

Programmation assembler



```
// delay function for 500ms
asm(
    "_delay:           \n\t"
    "LDI    r26, 255   \n\t"
    "LDI    r27, 12    \n\t"
    "LDI    r28, 50    \n\t"
    "_count:          \n\t"
    "DEC    r26         \n\t"
    "BRNE   _count     \n\t"
    "DEC    r27         \n\t"
    "BRNE   _count     \n\t"
    "DEC    r28         \n\t"
    "brne   _count     \n\t"
    "RET                    \n\t"
);

int main()
{
    // set LED port as output (DDRC = 0x14)
    asm("SBI    0x14, 2  \n\t");

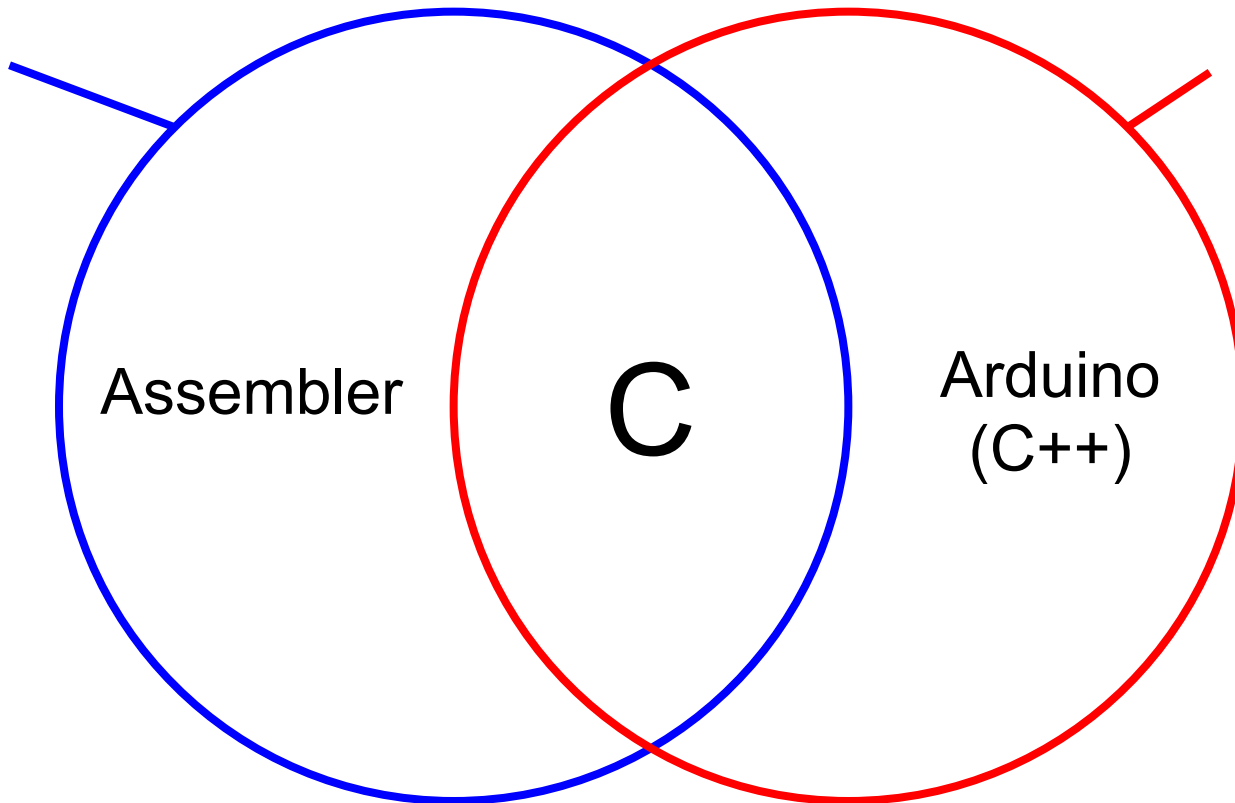
    // toggle LED (PORTC = 0x15) inside infinite loop
    asm(
        "_loop:\n\t"
        "SBI    0x15, 2   \n\t"
        "RCALL  _delay    \n\t"
        "CBI    0x15, 2   \n\t"
        "RCALL  _delay    \n\t"
        "RJMP  _loop     \n\t"
    );
}
```

Choix du langage



Taille
Vitesse

Lisibilité
Portabilité



Assembler

C

Arduino
(C++)

Action, réaction

```
int ir_sensor;

void setup()
{
  pinMode(PC(2), OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  ir_sensor = digitalRead(PA(0));
  if(ir_sensor == 0)
  {
    digitalWrite(PC(2), HIGH);
    Serial.write("Stop\n");
  }
  else
  {
    digitalWrite(PC(2), LOW);
    Serial.write("Go\n");
  }
  delay(500);
}
```



Action, réaction

```
int ir_sensor;
```

```
void setup()
{
  pinMode(PC(2), OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  ir_sensor = digitalRead(PA(0));
  if(ir_sensor == 0)
  {
    digitalWrite(PC(2), HIGH);
    Serial.write("Stop\n");
  }
  else
  {
    digitalWrite(PC(2), LOW);
    Serial.write("Go\n");
  }
  delay(500);
}
```





Types de variables: entiers

```
// nombres entiers signés
byte nb_8bit = 42; // -128 à 127
int8_t nb_8bit = 42; // -128 à 127

int nb_16bit = 42; // -32768 à 32767
int16_t nb_16bit = 42; // -32768 à 32767

// nombres entiers positifs
unsigned byte nb_u8bit = 42; // 0 à 255
uint8_t nb_u8bit = 42; // 0 à 255

unsigned int nb_u16bit = 42; // 0 à 65536
uint16_t nb_u16bit = 42; // 0 à 65536
```

Types de variables: à virgule



```
// nombres à virgule  
float nbv_32bit = 4.2;  
double nbv_64bit = 4.2;
```

Types de variables: énormes



```
// max: 4294967296 (4.3e9)
unsigned long int nb_32bit = 1234567890;
uint32_t nb_32bit = 1234567890;

// max: 1.844674407e19
uint64_t nb_64bit = 12345678901234567890;
```

Action, réaction



```
int ir_sensor;
```

```
void setup()
{
  pinMode(PC(2), OUTPUT);
  Serial.begin(9600);
}
```

```
void loop()
{
  ir_sensor = digitalRead(PA(0));
  if(ir_sensor == 0)
  {
    digitalWrite(PC(2), HIGH);
    Serial.write("Stop\n");
  }
  else
  {
    digitalWrite(PC(2), LOW);
    Serial.write("Go\n");
  }
  delay(500);
}
```


Action, réaction



```
int ir_sensor;

void setup()
{
  pinMode(PC(2), OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  ir_sensor = digitalRead(PA(0));
  if(ir_sensor == 0)
  {
    digitalWrite(PC(2), HIGH);
    Serial.write("Stop\n");
  }
  else
  {
    digitalWrite(PC(2), LOW);
    Serial.write("Go\n");
  }
  delay(500);
}
```

Action, réaction



```
int ir_sensor;

void setup()
{
  pinMode(PC(2), OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  ir_sensor = digitalRead(PA(0));
  if(ir_sensor == 0)
  {
    digitalWrite(PC(2), HIGH);
    Serial.write("Stop\n");
  }
  else
  {
    digitalWrite(PC(2), LOW);
    Serial.write("Go\n");
  }
  delay(500);
}
```

Action, réaction

```
int ir_sensor;

void setup()
{
  pinMode(PC(2), OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  ir_sensor = digitalRead(PA(0));
  if(ir_sensor == 0)
  {
    digitalWrite(PC(2), HIGH);
    Serial.write("Stop\n");
  }
  else
  {
    digitalWrite(PC(2), LOW);
    Serial.write("Go\n");
  }
  delay(500);
}
```



Action, réaction en C



```
#include <avr/io.h>
#include <util/delay.h>

#define BAUD 9600

#include <util/setbaud.h>

uint8_t ir_sensor, i;

void writeUart(const char *text)
{
    for(i = 0; text[i] != '\0'; i++)
    {
        // wait for empty buffer
        while((UCSRA & (1 << UDRE)) == 0);
        // set uart buffer value
        UDR = (unsigned char)text[i];
    }
}
```

```
int main()
{
    // set up uart communication
    UBRRH = UBRRH_VALUE;
    UBRRL = UBRRL_VALUE;
    UCSRA &= ~(1 << U2X);
    UCSRB |= 0x18;

    // set LED as output
    DDRC = 0b00000100;
    while(1)
    {
        // read pin 0 from PORTA
        ir_sensor = PINA & 0x00000001;
        if(ir_sensor == 0)
        {
            PORTC = 0b00000100;
            writeUart("Stop\n");
        }
        else
        {
            PORTC = 0;
            writeUart("Go\n");
        }
        _delay_ms(1000);
    }
}
```

GitHub



Infrastructures Photos R&D Wiki **GitHub**

EPFL



CALENDAR

0 - 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31



Robopoly
Robopoly

Lausanne

<http://robopoly.epfl.ch>

Joined on Jun 29, 2010

Repositories

Members

You are a member of this Organization!

All Public Private Sources Forks Mirrors



Arduino

C++ ★ 1 📌 0

Allows the PRisme kit to be programmed with Arduino IDE

Last updated 16 hours ago



PRisme2ISP

Arduino ★ 0 📌 0

Load programs directly using a second PRisme2 as an ISP programmer.

Last updated 17 hours ago



clock-7seg

Arduino ★ 1 📌 0

Makes a clock out of the 7 segment display. Needs the Arduino environment.

Last updated 3 months ago

La suite

Lundi 15/10/12

Démon moteurs, servomoteurs





FIN

Questions?