

# Systemes Embarques : Thermostat d'ambiance

Groupe B3G2H

Luc Rinaudo,  
Pierre Garcès,  
Sébastien Dupuy,  
Olivier Éloi.

Jun 2019

# Aspects pédagogiques

## Pré-requis

modèle d'architecture séquentielle :

constituants d'une architecture à microprocesseur, micro-contrôleur

## Objectifs pédagogiques

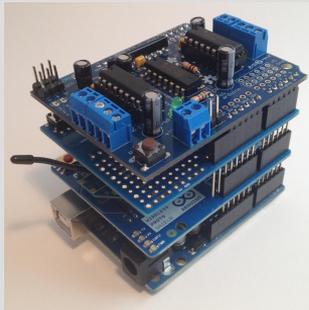
- Identifier le rôle des capteurs et actionneurs.
- Réaliser par programmation une IHM répondant à un cahier des charges donné.

# Mise en situation

Le thermostat d'ambiance confère un confort incomparable grâce à sa précision et sa réactivité tout en allégeant la facture d'énergie. Le thermostat que vous allez réaliser devra respecter les spécifications suivantes :

- être capable de mesurer la température d'ambiance avec une précision de  $0.1^{\circ}\text{C}$
- maintenir la consigne de température avec une précision de  $\pm 1^{\circ}\text{C}$ ,
- la consigne de température sera réglable à partir d'une télécommande infrarouge,
- l'utilisateur visualise les valeurs de la température actuelle, la température de consigne et l'état du système de chauffage (activé/désactivé) sur un afficheur LCD.

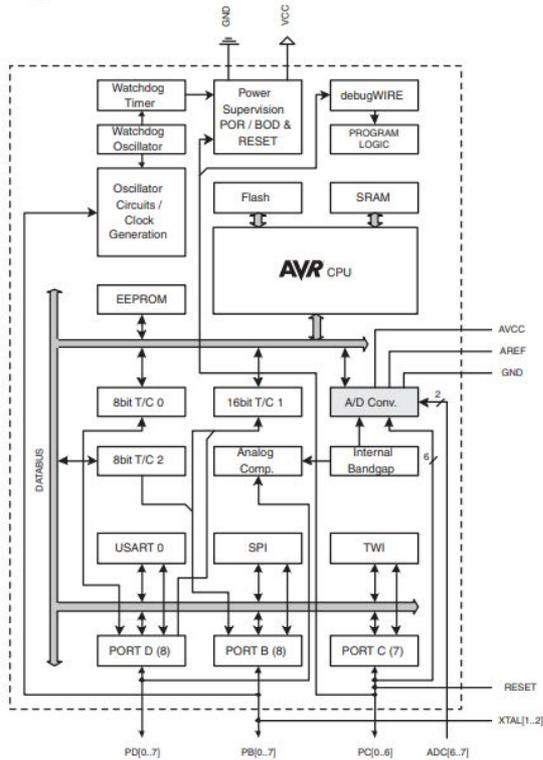
# Phase d'introduction : Arduino



- Chaîne de développement simplifiée,
- matériel Open Hardware, open Source
- évolutif (système de Shield)
- nombreuses bibliothèques pour gérer de nombreuses fonctionnalités et protocoles de communication (variateurs de vitesse, neoPixels, wifi, Ethernet, Lora...)
- Grande communauté d'utilisateurs et de développeurs...
- Nombreuses variantes (form factor)

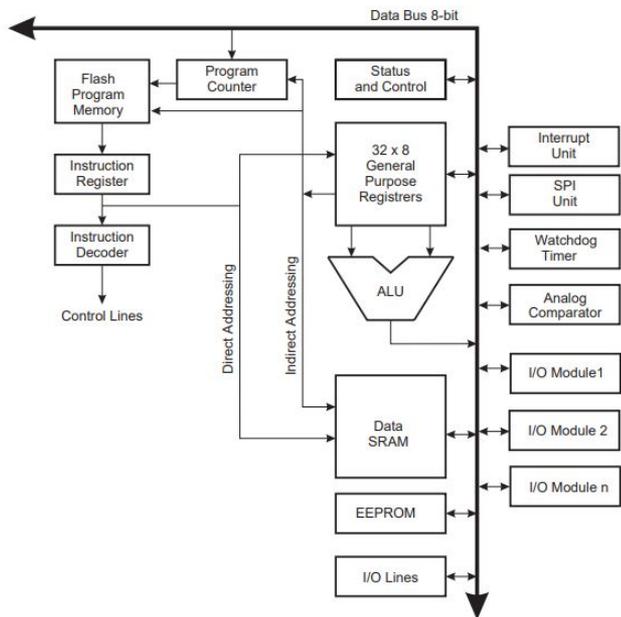
# Phase d'introduction : Arduino côté micro-contrôleur

Block Diagram



- Microprocesseur 8 bits - 20MHz
- matériel intégré pour communiquer suivant plusieurs protocoles 'bas niveau' : UART, I2C, SPI, TWI
- Port entrées/sorties standards (TTL-5V/3.3V)
- Convertisseur CAN / CNA(presque)
- Timers 16bits

# Phase d'introduction : Arduino côté micro-contrôleur



## Features

- High Performance, Low Power AVR<sup>®</sup> 8-Bit Microcontroller Family
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32KBytes of In-System Self-Programmable Flash program memory
  - 256/512/512/1KBytes EEPROM
  - 512/1K/1K/2KBytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security

# Activité élève : stratégie de développement

- Construction progressive du micrologiciel Thermostat,
- Chaque “brique” matérielle et logicielle est intégrée l’une après l’autre.

VOIR DOCUMENT ÉLÈVE

# Conclusion synthèse

## Les systèmes (informatiques) embarqués

### Les contraintes :

- ressources réduites (mémoire, calculs...), souvent gérées en temps réel,
- pas de système d'exploitation (ou Os spécifiques)
- ressources en énergie réduite (energy harvesting)

### Les avantages :

- flexibles (utilisable dans une large gamme d'applications),
- forte intégration des fonctions (pile TCP/IP, protocoles : BLE, Wifi, SigFox...)
- circuits mixtes (DSP, FPGA...),
- personnalisable (SOC)

# Conclusion synthèse (ter repetita)

## Préparation de l'environnement de travail :

Les bibliothèques (libraries) nécessaires à la gestion des différents périphériques doivent être installées de préférence avant (bien que cela puisse être fait par l'élève)

- DHT\_sensor\_library (Adafruit)
- Adafruit\_Sensor
- IRremote
- LiquidCrystal\_I2C

Les dernières versions de l'IDE Arduino possède un gestionnaire de bibliothèques (menu croquis\importer bibliothèque) qui permet de rechercher et installer automatiquement.

Elles sont également disponible pour une installation manuelle à l'adresse :

<https://www.arduino-libraries.info>

# Extension possibles

**Thermostat connecté (IOT)** : rajout d'un shield wifi ou module BLT pour interfaçage avec le téléphone, et/ou par interface web.

## Désassemblage d'un code simple :

En ligne de commande, taper :

```
avr_objdump -C -d simpleAddition.cpp.elf > simpleAddition.asm
```

```
simpleAddition.S
int a = 9;
void setup(){
  a = a + 1;
}
void loop(){
}
```

```
000000a6 <setup>: Récupération de la variable a (int sur 16 bits) donc opération en 2 temps
a6: 80 91 00 01    lds    r24, 0x0100    ; 0x800100 <__data_start>
aa: 90 91 01 01    lds    r25, 0x0101    ; 0x800101 <__data_start+0x1>
ae: 01 96         adiw   r24, 0x01      ; 1
b0: 90 93 01 01    sts    0x0101, r25    ; 0x800101 <__data_start+0x1>
b4: 80 93 00 01    sts    0x0100, r24    ; 0x800100 <__data_start>
b8: 08 95         ret
                                Stockage du résultat dans l'emplacement initial

000000ba <loop>:
ba: 08 95         ret

000000bc <main>:
bc: 0e 94 b5 00    call   0x16a    ; 0x16a <init>
c0: 0e 94 53 00    call   0xa6     ; 0xa6 <setup>
c4: c0 e0         ldi    r28, 0x00    ; 0
c6: d0 e0         ldi    r29, 0x00    ; 0
c8: 0e 94 5d 00    call   0xba     ; 0xba <loop>
cc: 20 97         sbiw   r28, 0x00    ; 0
ce: e1 f3         breq   .-8        ; 0xc8 <main+0xc>
d0: 0e 94 00 00    call   0         ; 0x0 <__vectors>
d4: f9 cf         rjmp  .-14       ; 0xc8 <main+0xc>
```

Addition du contenu de R24 avec la valeur immédiate '1' : résultat stocké dans R24 lui-même

# Système de connexion Grove

- pas de câblage hasardeux
- de nombreux modules disponibles (capteur, RTC, boutons, LEDS, afficheurs...)
- modules ré-utilisable pour la RaspberryPi, BBC micro:bit

<https://www.gotronic.fr/cat-modules-grove-1361.htm>

