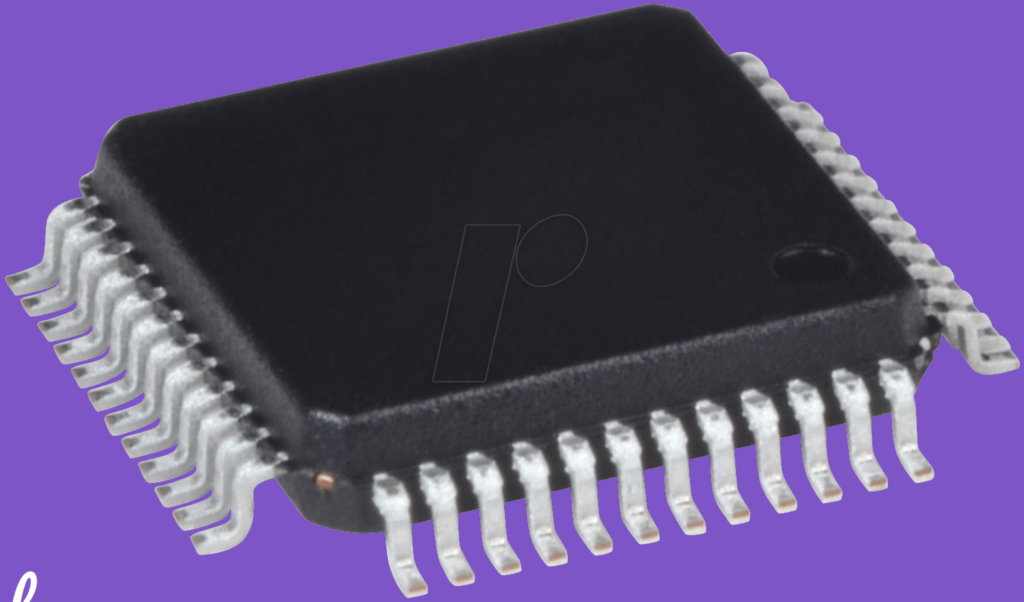
Aalto-yliopisto
Sähkötekniikan
korkeakoulu



The secrets of Microcontrollers

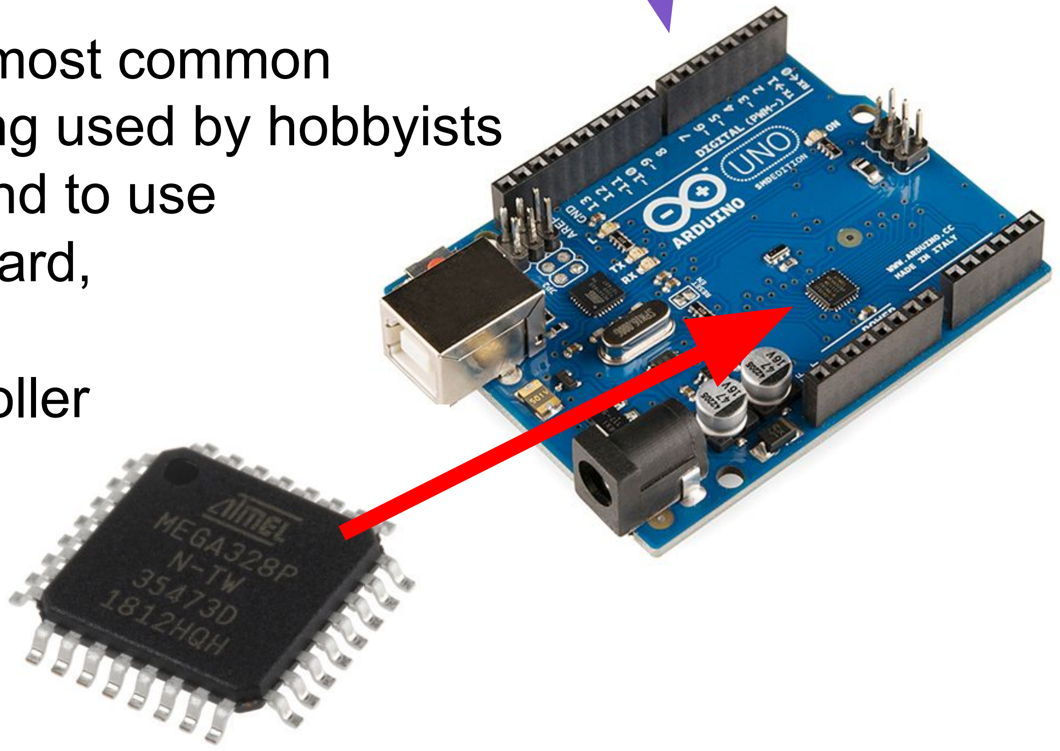
Hey hey, ho ho, 1100110!

Microcontrollers (MCUs)

- Microcontroller is an integrated circuit (IC) that contains at least a processor and some type of memory (Flash, RAM, ROM…). Usually they also have some peripherals like Analog-to-Digital Converter (ADC), UART…
- Usually programmed to run a single program with various functions
- Can also be programmed to have an operating system (like FreeRTOS) to run multiple programs ‘simultaneously’

Arduino

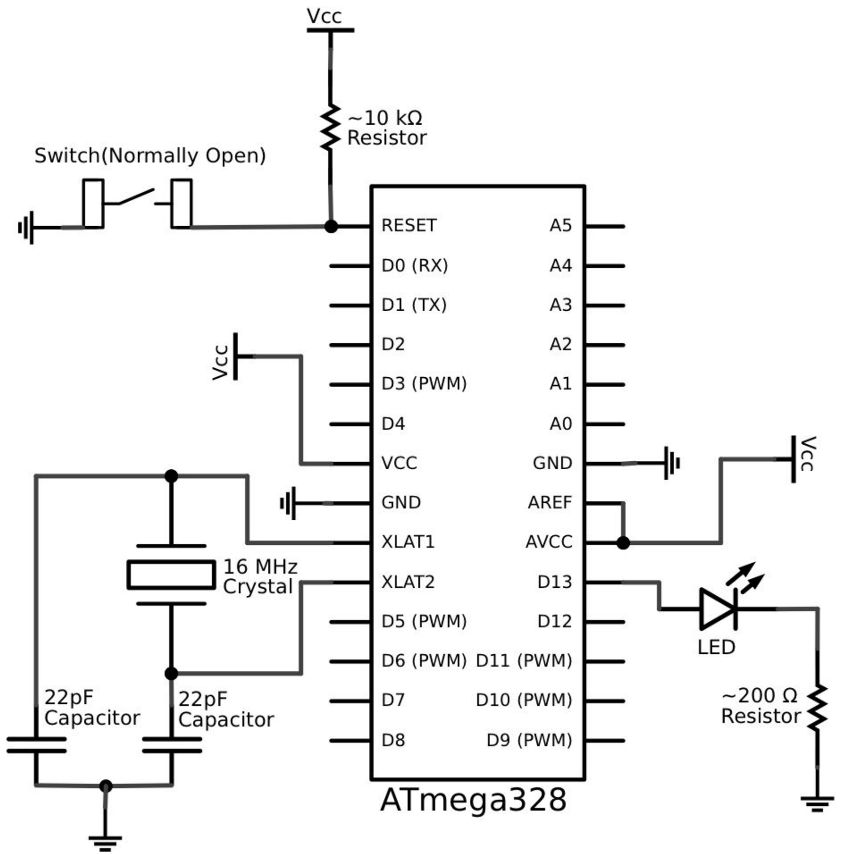
- Arduino Uno is one of the most common microcontroller boards being used by hobbyists
- Simple, easy to program and to use
- Arduino is development board, not microcontroller
- Atmega328p is microcontroller



Aalto-yliopisto
Sähkötekniikan korkeakoulu

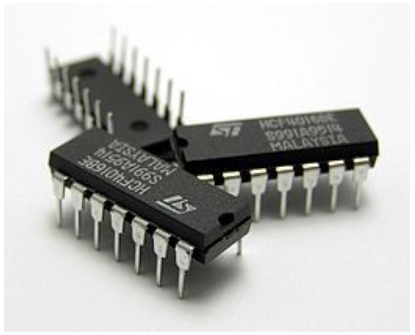
Making own Arduino/MCU board?

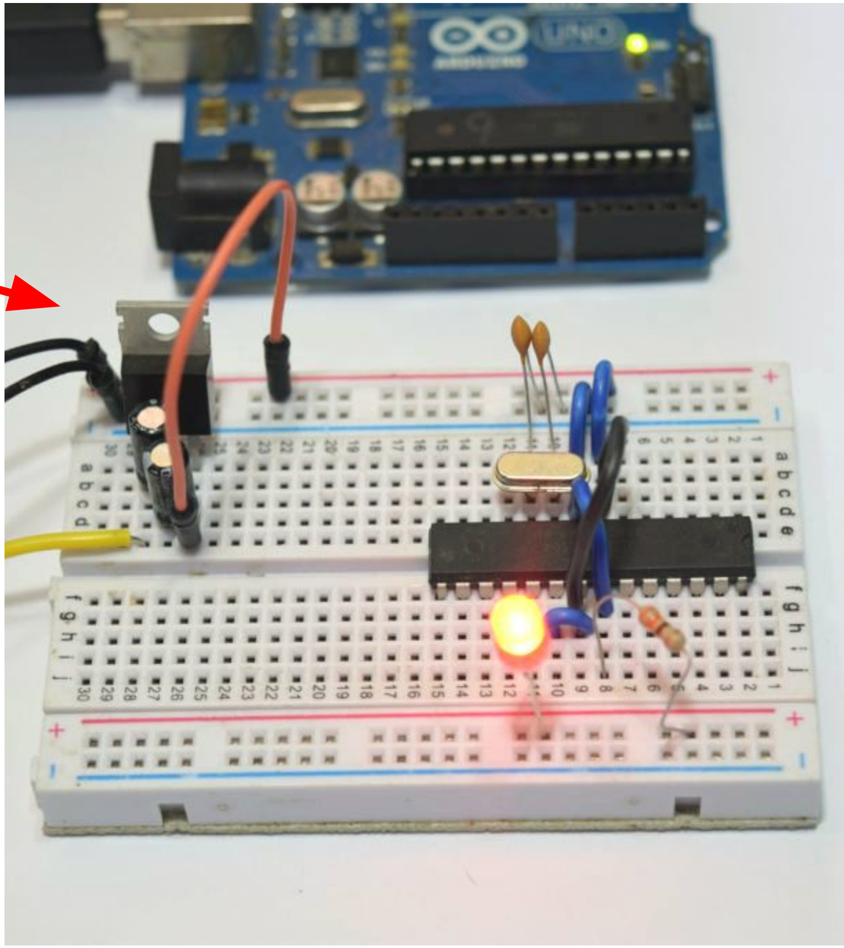
- You can copy the design from Arduino schematic.
- Microcontroller datasheet explains the function of all the pins on the microcontroller.
- You can design a PCB with KiCad for example.



Making own Arduino

- If the microcontroller is in THT package e.g. dual inline package, you can use it on a breadboard!





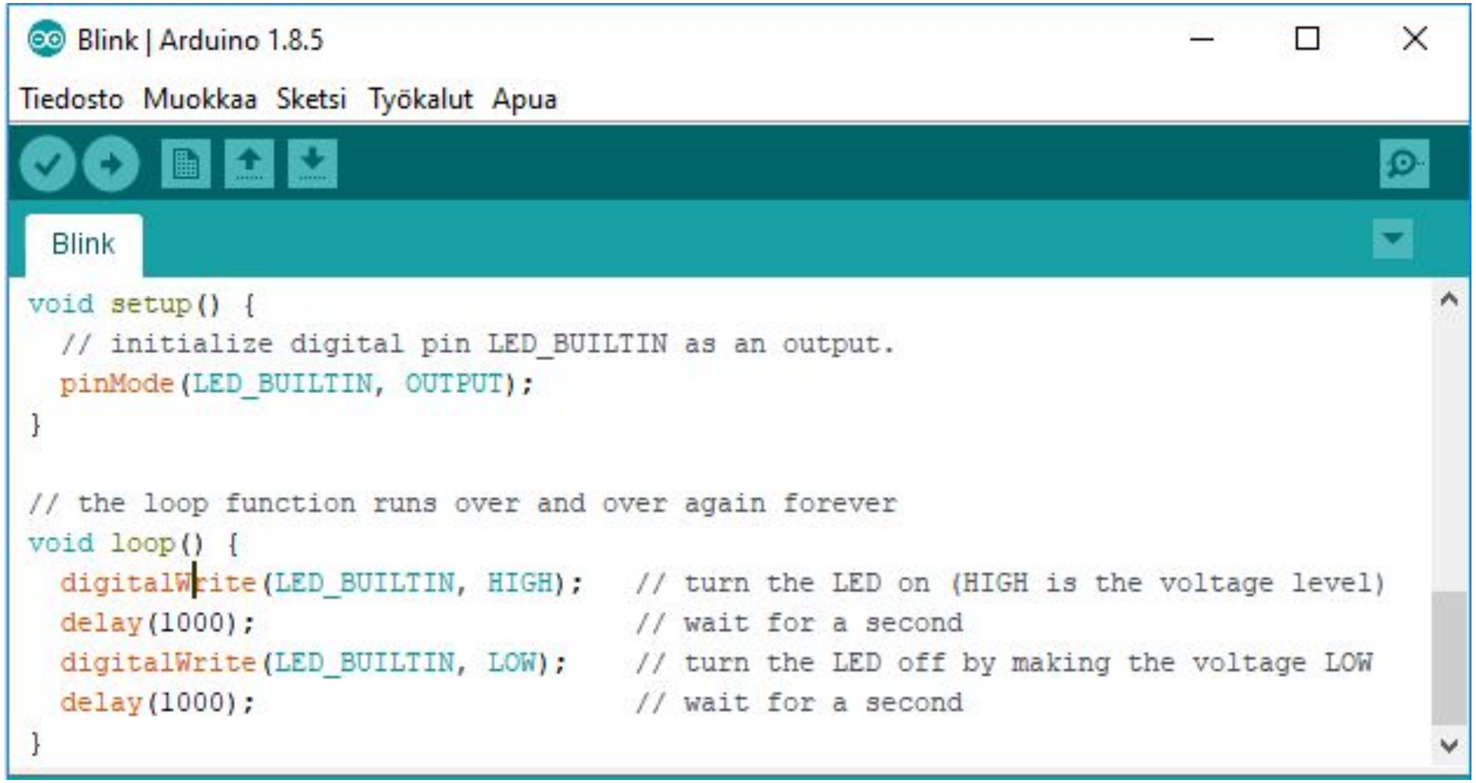
- To program it you need to use another Arduino, or a specialized programmer
- You can find instructions with google

Aalto-yliopisto
Sähkötekniikan korkeakoulu

Programming with C

- The easiest way => Arduino IDE
  - Program is written in Arduino language; based on C++ (and C-language)
  - Easier than programming directly using MCU's register - MCU is controlled by registers, switches
  - With Arduino, user calls for functions that take care of registers automatically

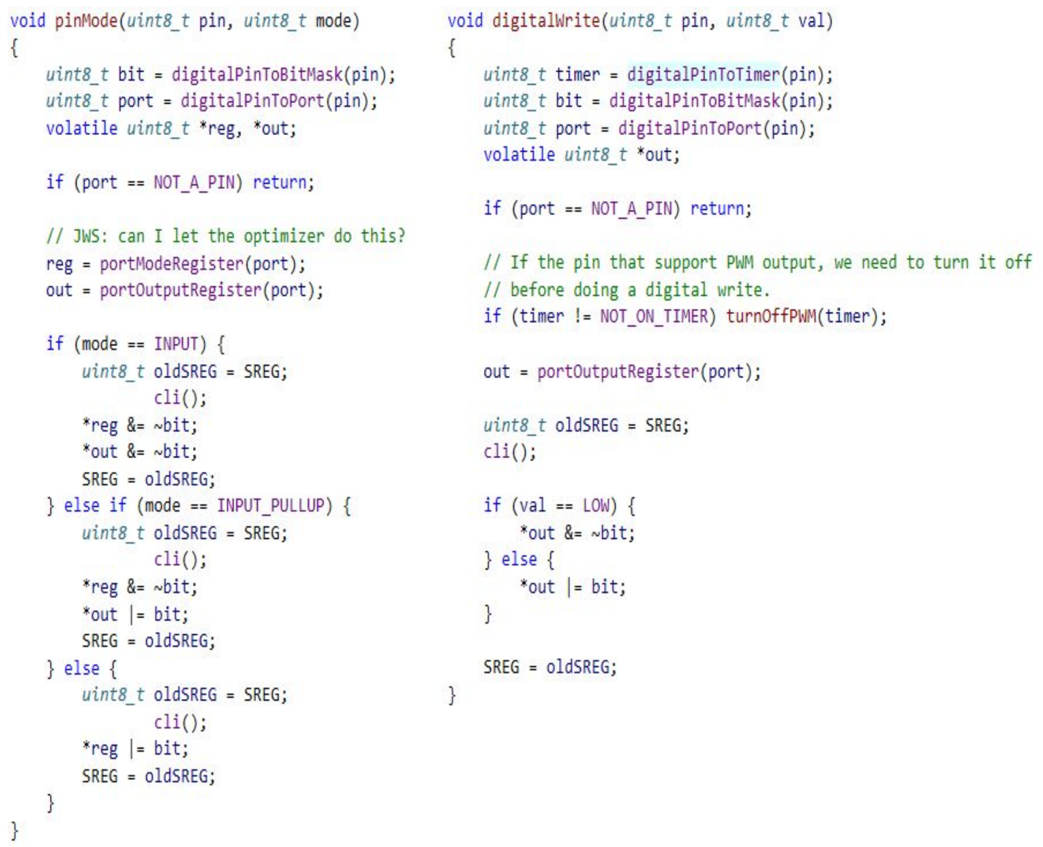
Simple code to blink an LED



```
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

But under the hood

- Functions after functions after functions…
- With time critical applications this becomes an issue...



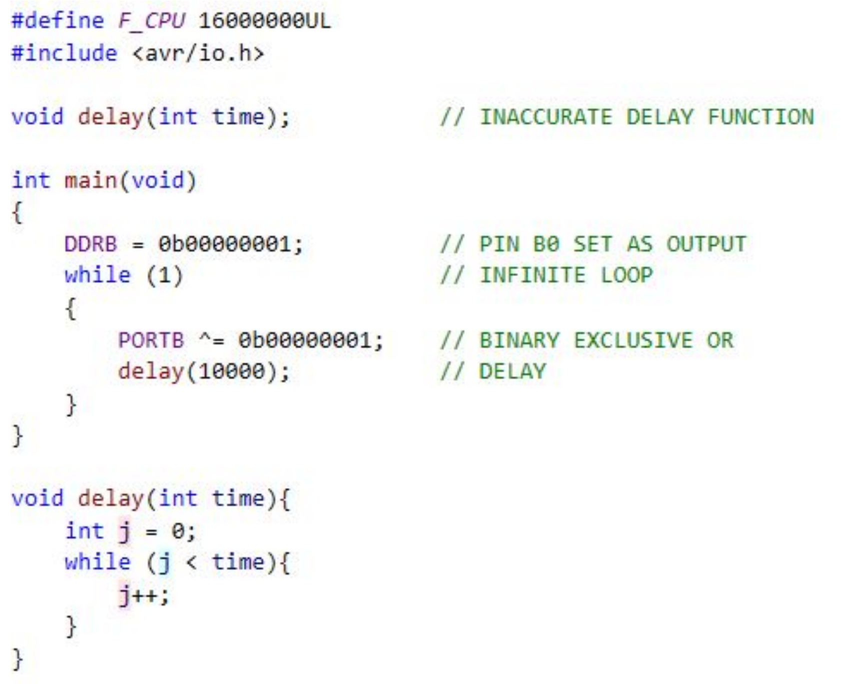
```
void pinMode(uint8_t pin, uint8_t mode)
{
    uint8_t bit = digitalPinToBitMask(pin);
    uint8_t port = digitalPinToPort(pin);
    volatile uint8_t *reg, *out;

    if (port == NOT_A_PIN) return;

    // JWS: can I let the optimizer do this?
    reg = portModeRegister(port);
    out = portOutputRegister(port);

    if (mode == INPUT) {
        uint8_t oldSREG = SREG;
                cli();
        *reg &= ~bit;
        *out &= ~bit;
        SREG = oldSREG;
    } else if (mode == INPUT_PULLUP) {
        uint8_t oldSREG = SREG;
                cli();
        *reg &= ~bit;
        *out |= bit;
        SREG = oldSREG;
    } else {
        uint8_t oldSREG = SREG;
                cli();
        *reg |= bit;
        SREG = oldSREG;
    }
}
```

```
void digitalWrite(uint8_t pin, uint8_t val)
{
    uint8_t timer = digitalPinToTimer(pin);
    uint8_t bit = digitalPinToBitMask(pin);
    uint8_t port = digitalPinToPort(pin);
    volatile uint8_t *out;

    if (port == NOT_A_PIN) return;

    // If the pin that support PWM output, we need to turn it off
    // before doing a digital write.
    if (timer != NOT_ON_TIMER) turnOffPWM(timer);

    out = portOutputRegister(port);

    uint8_t oldSREG = SREG;
    cli();

    if (val == LOW) {
        *out &= ~bit;
    } else {
        *out |= bit;
    }

    SREG = oldSREG;
}
```

Aalto-yliopisto
Sähkötekniikan korkeakoulu

Register code version

- With direct register code the same task is performed with fewer clock cycles!
- More difficult to write and read but occupies less memory and executes faster



```c
#define F_CPU 16000000UL
#include <avr/io.h>

void delay(int time);           // INACCURATE DELAY FUNCTION

int main(void)
{
    DDRB = 0b00000001;          // PIN B0 SET AS OUTPUT
    while (1)                   // INFINITE LOOP
    {
        PORTB ^= 0b00000001;    // BINARY EXCLUSIVE OR
        delay(10000);           // DELAY
    }
}

void delay(int time){
    int j = 0;
    while (j < time){
        j++;
    }
}
```

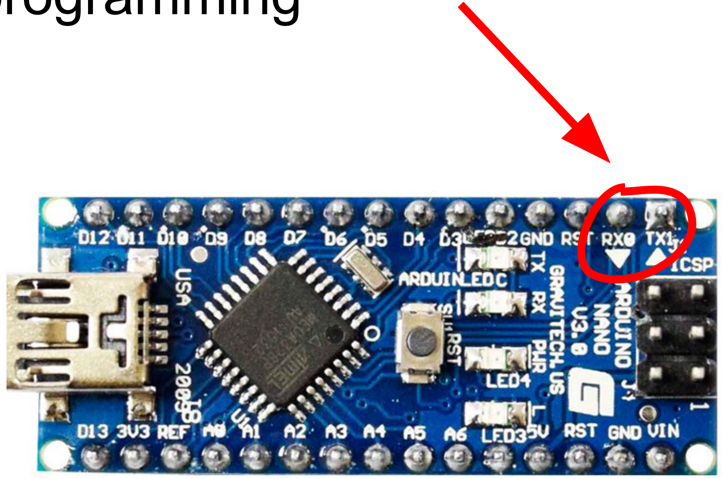
Basic Arduino functions

- Most common functions:
  - `pinMode(pin, mode)`
  - `digitalWrite(pin, value)`
  - `digitalRead(pin)`
  - `analogWrite(pin, duty_cycle) [0 - 255]`
  - `analogRead(pin) [0 - 1023]`

- And more functions can be found from arduino.cc

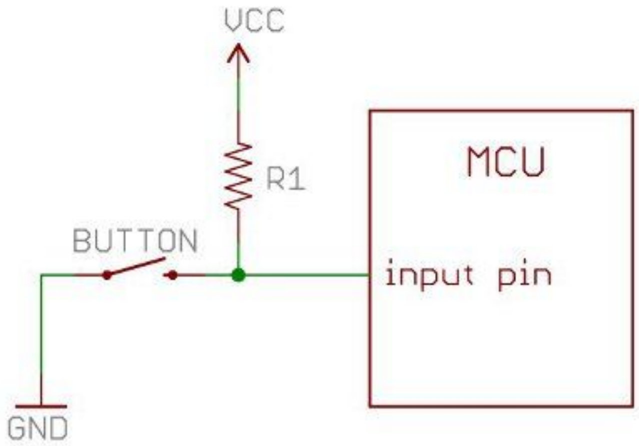
Common mistakes

- Using pins 0 and 1 will block UART programming



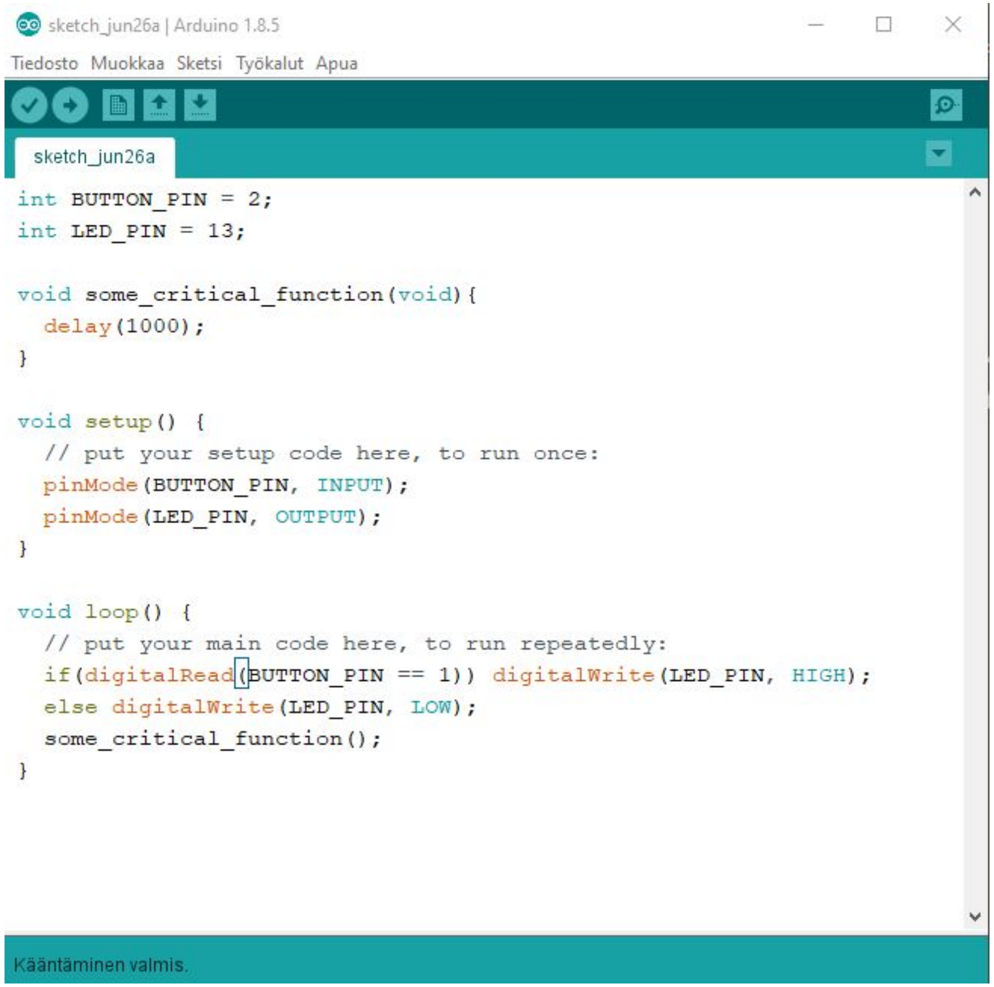
D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RX0 TX1
ICSP
ARDUINO NANO V3.0
D13 3V3 REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN

- Floating pins



Interrupts

- Let’s take a look at this example code:
- The problem is polling.


sketch_jun26a | Arduino 1.8.5
Tiedosto Muokkaa Sketsi Työkalut Apua
sketch_jun26a

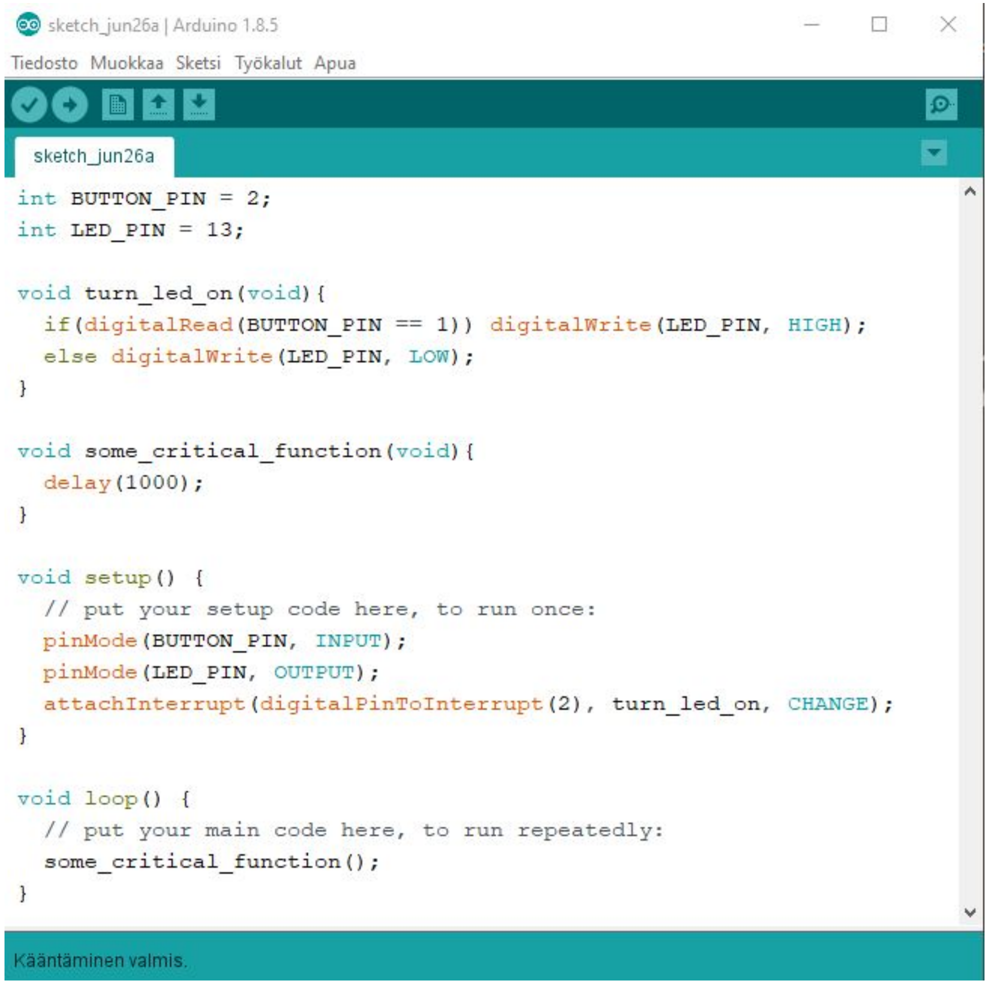
```
int BUTTON_PIN = 2;
int LED_PIN = 13;

void some_critical_function(void){
  delay(1000);
}

void setup() {
  // put your setup code here, to run once:
  pinMode(BUTTON_PIN, INPUT);
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  if(digitalRead(BUTTON_PIN == 1)) digitalWrite(LED_PIN, HIGH);
  else digitalWrite(LED_PIN, LOW);
  some_critical_function();
}
```

Kääntäminen valmis.

Aalto-yliopisto
Sähkötekniikan korkeakoulu

Interrupts

- Solution: Interrupts!


sketch_jun26a | Arduino 1.8.5
Tiedosto Muokkaa Sketsi Työkalut Apua
sketch_jun26a

```
int BUTTON_PIN = 2;
int LED_PIN = 13;

void turn_led_on(void){
  if(digitalRead(BUTTON_PIN == 1)) digitalWrite(LED_PIN, HIGH);
  else digitalWrite(LED_PIN, LOW);
}

void some_critical_function(void){
  delay(1000);
}

void setup() {
  // put your setup code here, to run once:
  pinMode(BUTTON_PIN, INPUT);
  pinMode(LED_PIN, OUTPUT);
  attachInterrupt(digitalPinToInterrupt(2), turn_led_on, CHANGE);
}

void loop() {
  // put your main code here, to run repeatedly:
  some_critical_function();
}
```

Kääntäminen valmis.

Aalto-yliopisto
Sähkötekniikan korkeakoulu

Interrupts

- Interrupts can be set to react on all kinds of signals!
    - ADC Conversion ready!
        - *analogRead(channel) is polling!*
    - UART has received a byte!
    - External interrupts!
    - Timer interrupts!
    - ...

Timers

- Timers are clocks that count ticks from the prosessor
- Arduino Uno has three timers:
  - Two 8-bit timer (can count from 0 to 255)

    and one 16-bit timer (count from 0 to 65535)
- Can measure time accurately.
- Arduino has two functions to measure time:
  - `unsigned long time = millis();`
  - `unsigned long time = micros();`

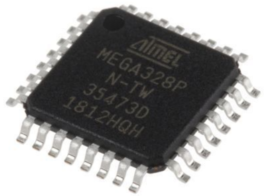
Choosing an MCU

- **Which buses needed?**
  - UART
  - SPI
  - I2C
  - USB
  - Ethernet
  - CAN

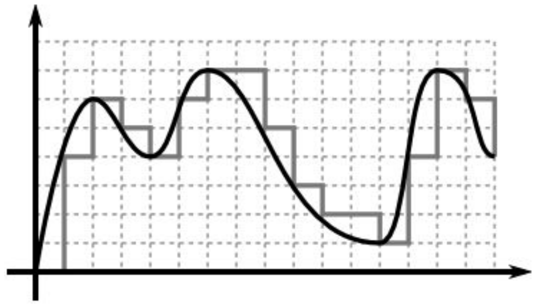
- **Wireless connectivity?**
  - WiFi
  - Bluetooth

- **How many pins needed?**
- **Easy to program?**
- **Enough memory?**
- **Clock speed MHz?**

- **Package?**
  - DIP
  - TQFP



- **ADC**
- **DAC**
- **PWM**



Where to buy microcontrollers?

- **mouser**
- **tme**
- **farnell**
- **digikey**
- and many other online stores