



IoT OG EMBEDDEDE SYSTEMER 1

- Med Lærke Brandhøj Kristensen
- lkri@techcollege.dk

IoT faget

- Hardware
 - Arduino
 - chipsæt
 - komponenter
- Software
 - Programmering med microcontroller
 - C/C++
 - interrupts
- Kommunikations teknologier
 - Serial
 - I2C
 - Wifi
 - Zigbee

ATMega xx Assembler Instruktionser

- ATMega xx Assembler instruktioner, inddeles i 5 hovedgrupper.
- 1) Aritmetiske og logiske instruktioner.
- 2) Hop (Branch) instruktioner.
- 3) Bit og Bit Test instruktioner.
- 4) Data Transfer instruktioner.
- 5) Microcontroller (MCU) instruktioner.
- Alle assembler instruktionerne i de forskellige hovedgrupper kan findes i databladet for den processor, man arbejder med.

ATMega xx, Statusregister

The 8-bit status register contains these bits:

7	6	5	4	3	2	1	0
I	T	H	S	V	N	Z	C

- . I: interrupt enable/disable flag
- . T: transfer bit for BLD and BST instructions
- . H: half carry - useful for decimal arithmetic
- . S: signed tests
- . V: two's complement overflow
- . N: set for if result < 0
- . Z: set for if result $== 0$
- . C: useful for multi-byte operations

Basale af C++ for Arduino

- Blink eksempel

```
// the setup function runs once when you press reset or power the board
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
}
```

<https://docs.arduino.cc/built-in-examples/>

Digital Read

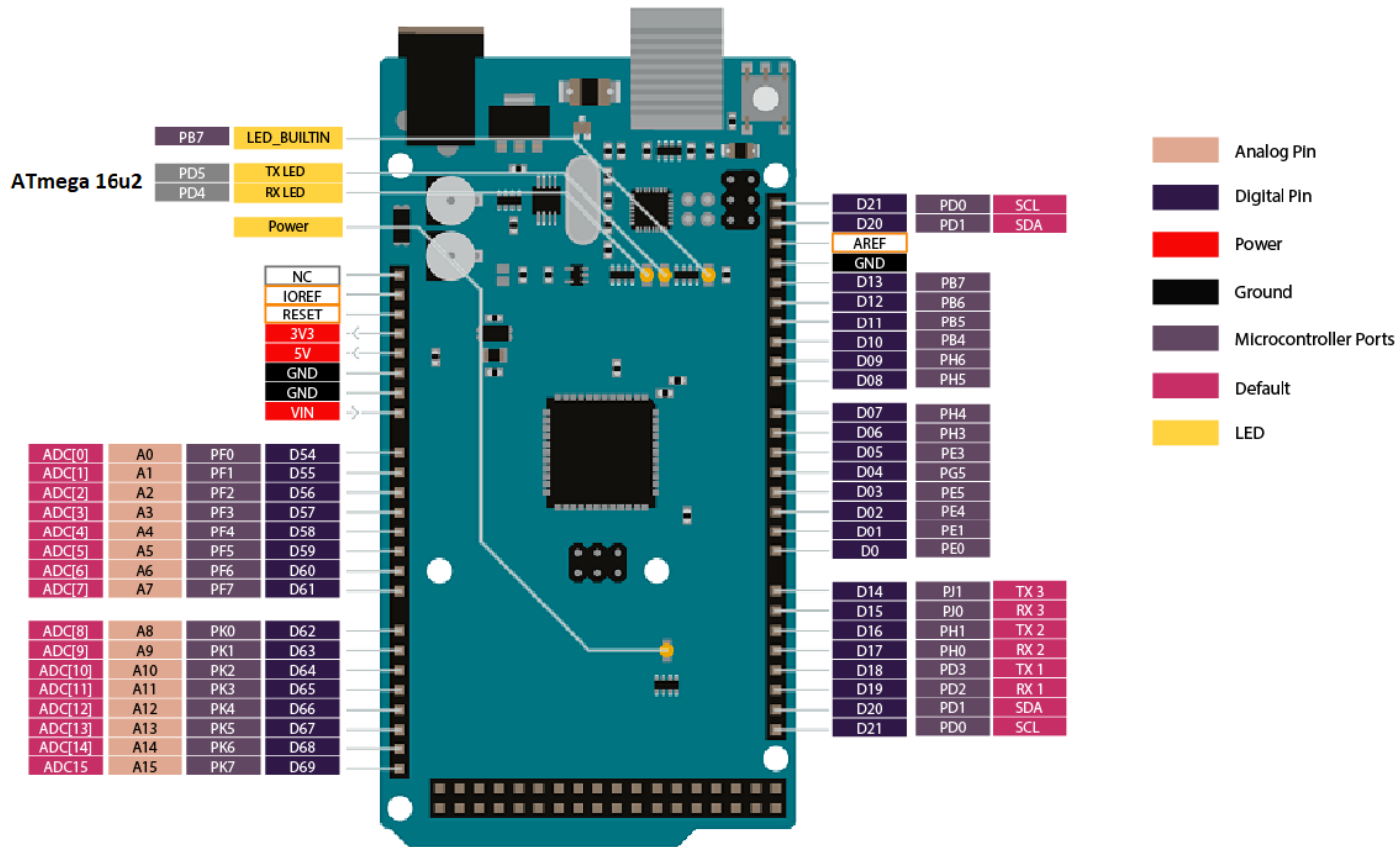
```
// digital pin 2 has a pushbutton attached to it. Give it a name:
int pushButton = 2;

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input:
  pinMode(pushButton, INPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input pin:
  int buttonState = digitalRead(pushButton);
  // print out the state of the button:
  Serial.println(buttonState);
  delay(1);          // delay in between reads for stability
}
```

Arduino Mega - Pinout

www.theengineeringprojects.com



Arduino Mega 2560 Rev3 Pinout

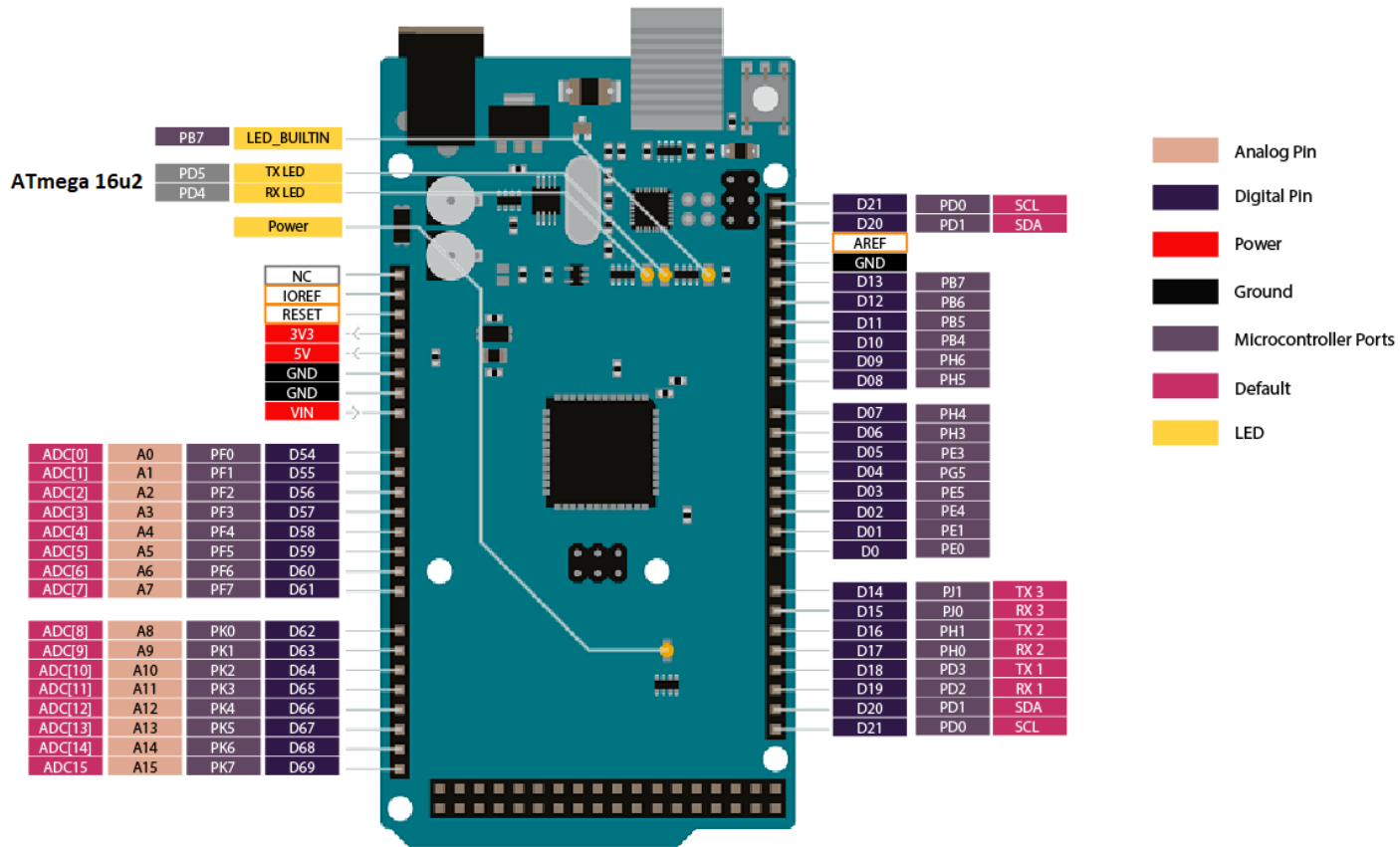
Analog read

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);          // delay in between reads for stability
}
```

Arduino Mega - Pinout

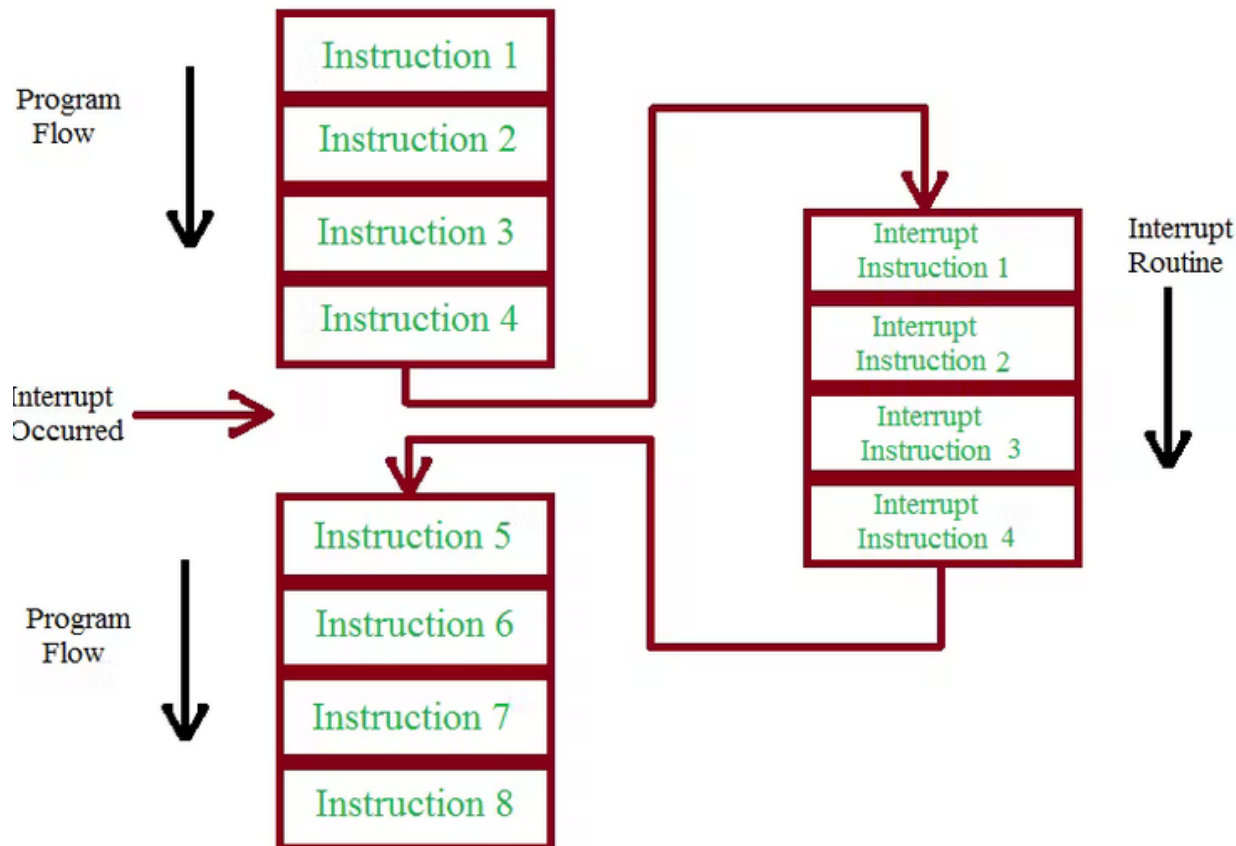
www.theengineeringprojects.com



Arduino Mega 2560 Rev3 Pinout

Interrupts

- Interrupts stopper det normale program flow
- Det kan lade sig gøre med interne og eksterne
 - Eksterne/Hardware interrupts: Pin change
 - Interne/Software interrupts: Instruktioner



Interrupts C++

```
int x=0, y=0;           //some variables to increment
int ledToggle = LOW;    //led state
//variables to keep track of the timing of recent interrupts
unsigned long button_time = 0;
unsigned long last_button_time = 0;
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
  Serial.begin(9600);
  attachInterrupt(digitalPinToInterrupt(buttonPin), button_ISR, CHANGE);
}
void loop(){
  Serial.print("loop ");
  Serial.print(x++ );
  Serial.println();
  delay(500);
}
void button_ISR(){
  button_time = millis();
  //check to see if increment() was called in the last 250 milliseconds
  if (button_time - last_button_time > 250){
    Serial.print("Interrupt ");
    Serial.print(y++);
    Serial.println();
    ledToggle = !ledToggle;
    digitalWrite(ledPin, ledToggle);
    last_button_time = button_time;
  }
}
```

