

Organization

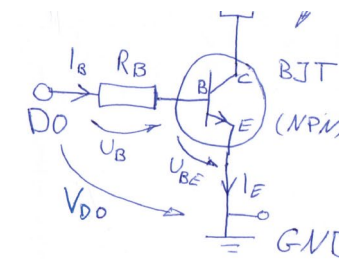
Lecturers	Prof. Dr. Wolfgang Grote-Ramm M.Sc. Jörn Hornig
Lecture (2 SWS) and Hands-on exercises (2 SWS)	Tuesdays 9:00 – 12:30 weekly lab 05.2.018
Workload	180h 60h (attendance time) 120h (self-study)
Credits	6 CP
Conditions of participation	Nothing special. Basic Programming skills warmly recommended (e.g., C, Java, Matlab).
Type of examination	Homework (50 %) and oral exam (50 %)
Language	English

General Information

The elective strongly focusses on practical exercises using a microcontroller (μC) evaluation board (Arduino) for solving typical automation tasks by connecting the μC to the real world (sensors, actuators, electronic devices, LCD panels ...) and the internet (IoT).

Students will program their μC in the lab and at home using the Arduino IDE (and Matlab/Simulink for some occasions) with their own laptops. Some existing programming skills (C, Java or Matlab) are warmly recommended for this course. Evaluation board and other required equipment will be lent and handed over to the students in the first session. Alternatively, students can buy their own equipment (approx. 75 €).

The number of participants is limited due to limited equipment.



U_{BE} : Voltage drop

$$I_B \cdot h_{FE} = I_C \quad h_{FE} \gg 1 : \text{Transistor gain} \rightarrow \text{data sheet}$$

Selecting R_B :

$$V_{D0} = U_{BE} + U_{R_B} = U_{BE} + R_B \cdot I_B$$

$$V_{D0} = R_B \cdot \frac{I_C}{h_{FE}} + U_{BE}$$

Contents

- Range of application for μC , μC architecture, programming of μC , basics of C programming and cross compilation.
- Real-time systems, I/O ports, system clock, timers and interrupts, timing in automation and control tasks with time-discrete systems
- Basic electronics in the peripheral μC -context, acquisition of analog data (ADC), output of analog data (DAC, PWM), communication via interfaces (serial bus, I²C, SPI), μC as embedded system, networking with μC s, connection and circuitry of actuators and sensors
- (Industrial) Internet of Things
- Rapid prototyping of automation algorithms on μC using Matlab/Simulink.
- Most subjects are elaborated practically with hands-on exercises in the context of automation and open and closed loop control.

```

Blink
// include header files for libraries
#include <Arduino.h> // not used in this example
// defines
#define DELAYTIME 1000 // milliseconds

// global variables
char text[40] = "Hello world, here I am ...";

// 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);
  Serial.begin(9600);
  // serial connection initialization:
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only.
  }
  // prints title with ending line break
  Serial.println("Serial connection established...");
}

// the loop function runs over and over again forever
void loop() {
  // the Arduino "main function"
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  Serial.println(text); // print variable text to serial connection
  delay(DELAYTIME); // wait for DELAYTIME ms
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(DELAYTIME); // wait for DELAYTIME ms
}
  
```

