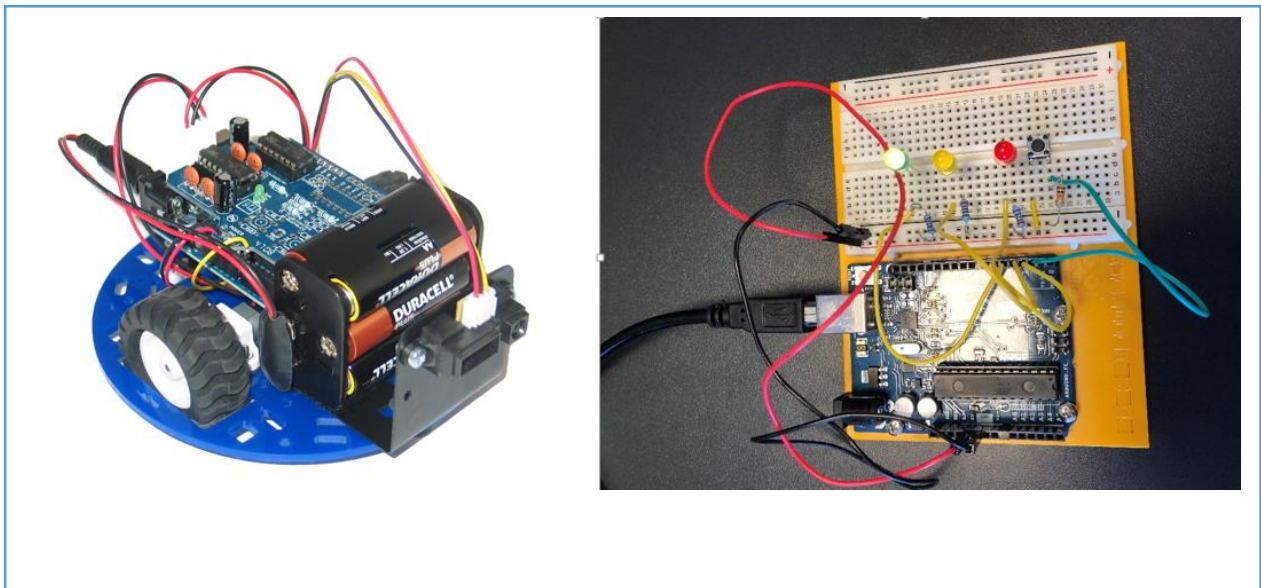




ARDUINO & ARDUINO ROBOT

Trafiklys og Robot



DIKSHYA SINGH SHAH

JUNE 28, 2025

TEC-BALLERUP
Telegrafvej-9

Table of Contents

Indledning.....	2
Beskrivelse af Arduino projektet	2
Øvelse 1. Trafiklys:.....	2
Nødvendigt materialer for trafiklys:.....	2
Hardware udstyr til trafiklys:	2
Software til afvikling:	2
Endelige billede af trafiklys.....	3
Beskrivelse af Trafiklys kode.....	3
Billede af trafiklys kode	3
Billede af Arduino GUI trafiklys kode	4
Øvelse 2. Robot (Del 1)	5
Øvelsen 3. Robot (Del 2):.....	6
Nødvendigt materialer for Robot	6
Hardware udstyr til øvelsen:	6
Software til afvikling:	6
Beskrivelse af Arduino Robot kode	6
Billede af Arduino Robot Kode.....	7
Source Kode med kommentarer	8
Kode af traffic Lys fra arduino projekts bog:	8
Kode af TrafikLys:.....	9
Kode af Robot	11
Konklusion:	13

Indledning

Dette Arduino-projekt forklarer, hvorfor det er en god idé at starte med at løse simple projekter – for eksempel at få en LED-diode til at blinke. Øvelserne i Arduino Projects Book skal udføres og dokumenteres.

Beskrivelse af Arduino projektet

Arduino er en open source-elektronikplatform baseret på brugervenlig hardware og software. Et Arduino-board er et printet kredsløb (PCB), designet til at bruge en mikrocontroller-chip samt forskellige input- og output-komponenter.

Hvert Arduino-program har to hovedfunktioner, som er dele af computerprogrammet. Disse funktioner udfører specifikke kommandoer, har unikke navne og kaldes, når det er nødvendigt. De nødvendige funktioner – setup og loop – skal deklarereres. Det betyder, at vi skal fortælle Arduinoen, hvad disse funktioner skal gøre.

Øvelse 1. Trafiklys:

Nødvendigt materialer for trafiklys:

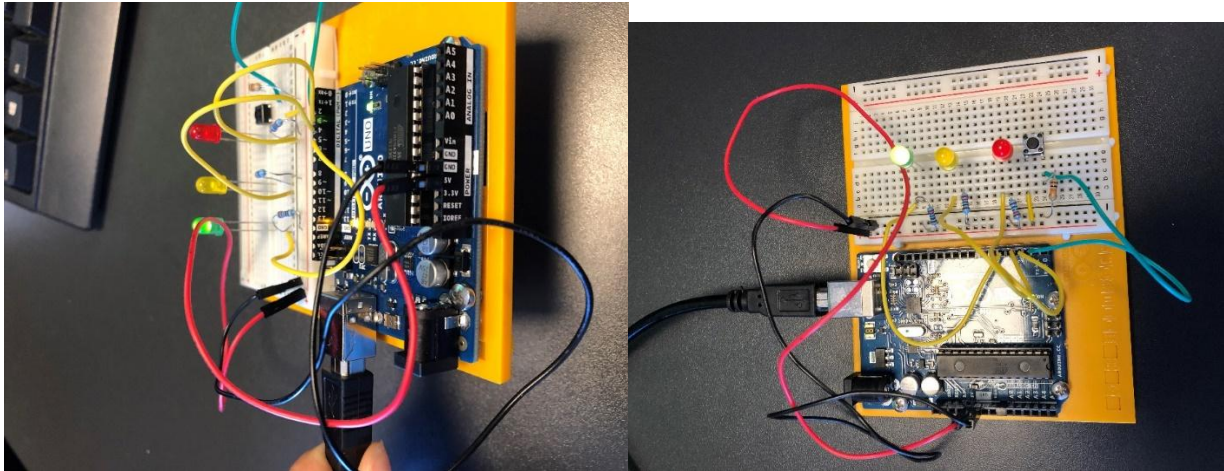
Hardware udstyr til trafiklys:

- Arduino startkit
- Arduino Projects Book

Software til afvikling:

1. C++ Compiler
2. Arduino IDE

Endelige billede af trafiklys: sådan ser ud det færdige trafiklys ud:



Beskrivelse af Trafiklys kode

Dette er koden til at styre et trafiklys. Først har jeg brugt en integer til at gemme hele tallet for switchState. Ved hjælp af pinMode har jeg konfigureret digitale pins som input og output. Jeg har også brugt if og else til at håndtere, om trykknappen er trykket eller ej. Desuden har jeg angivet, hvor lang tid systemet skal vente, hvis knappen ikke trykkes.

Billede af trafiklys kode

```
Dikshya | Arduino 1.8.8
File Edit Sketch Tools Help

Dikshya

int switchstate = 0;

void setup() {
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(2, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  switchstate = digitalRead(2);

  if (switchstate == LOW) {
    // the button is not pressed
    digitalWrite (3,HIGH); //green Led
    digitalWrite (4,LOW); //red Led
    digitalWrite (5,LOW); //red Led
  }
  else{ //the button is pressed
    digitalWrite (3,LOW);
    digitalWrite (4,LOW);
    digitalWrite (5,HIGH);
  }
  delay (250); // wait for a quarter second
  //toggle the leds
  digitalWrite(4,LOW);
  digitalWrite(5, LOW);
  delay(250); // wait for a quarter second
} // go back to the beginning of the loop
```

Billede af Arduino GUI trafiklys kode

Når installationsprocessen er fuldført vil man blive mødt af Arduino IDE'en (brugerfladen) som ser sådan her ud:



```
anothercodefor_trafik_lys | Arduino 1.8.8
File Edit Sketch Tools Help

anothercodefor_trafik_lys

int switchstate = 0;
int light = 0;

int pinnum[] = {5,4,3};

void setup() {
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(2, INPUT);
}

void loop() {
  while (true) {
    int presscontrol = 0;

    for (int i = 0; i < pinnum.length; i++) {
      digitalWrite(pinnum[i], HIGH);
      delay(1000);

      light = digitalRead(3);
      switchstate = digitalRead(2);

      digitalWrite(pinnum[i], LOW);

      if (switchstate == HIGH && presscontrol == 0) {
        presscontrol = 1;
      }

      if (presscontrol == 1 && light == HIGH) {
        delay(1000);
        digitalWrite(pinnum[i], LOW);

        delay(1000);

        digitalWrite(5, HIGH);
        delay(3000);
        digitalWrite(5, LOW);
      }
      else {
      }
    }
  }
}
```

Donna Savino

Før man begynder at kode og uploade sit script til Arduino boardet, skal man huske at vælge den rigtige port og board -type, så IDE'en ved, hvor den skal sende koden hen.

For at vælge port :

Tools -> Port, og vælg derefter den ønskede port.

Øvelse 2. Robot (Del 1)

Hvad kan robotteknologien bruges til i fremtiden?

Fremtiden for robotteknologi er lovende. Robotter vil kunne observere, udføre opgaver, kommunikere og besidde avancerede evner og intelligens.

Hvordan bruges robotteknologien lige nu?

Robotter findes allerede i mange hjem. De hjælper med huslige opgaver, minder os om vores tidsplan og kan endda underholde børn. Et kendt eksempel er den autonome støvsuger Roomba.

Hvad mener I at fremtiden er for samarbejdet mellem programmører og robotteknologien?

Robotter fungerer ikke uden programmering. Kombinationen af datavidenskab og robotteknologi fører til innovation som selvkørende biler. Virksomheder som Tesla, Ford, Volkswagen og BMW arbejder alle på næste generation af transport, hvor passagerer blot skal læne sig tilbage og nyde turen.

Forklar hvad en 'EMORO 2560 CONTROLLER' er og hvordan det kan bruges i robotteknologien?

EMoRo 2560 er den første Arduino-kompatible controller med CE-certificering. Den er bygget op omkring en ATmega2560-mikrocontroller og har mange tilslutningsmuligheder.

Den programmeres med Arduino IDE, og der findes biblioteker og eksempler, som kan hentes via Library Manager. Når koden er klar, kan den uploades til EMoRo 2560 via USB Mini, JTAG-10 eller ISP-6. Den bruger en Arduino-kompatibel bootloader, og kræver derfor ikke særlige programmer eller drivere.

Øvelsen 3. Robot (Del 2):

Nødvendigt materialer for Robot

Hardware udstyr til øvelsen:

1. 9V batteri / strømforsyning
2. Arduino robot

Software til afvikling:

1. C++ Compiler
2. Arduino IDE

Beskrivelse af Arduino Robot kode

Sådan lavede jeg et program til robotten, som kan undgå forhindringer ud fra en planlagt rute.

Jeg startede med at definere værdierne for hver motor i Arduino-programmet. For at venstre hjul skulle bevæge sig fremad, gav jeg det værdien 500, så det kunne rotere med uret. For at bevæge sig baglæns gav jeg det værdien 2500, så det kunne rotere mod uret.

For højre hjul brugte jeg værdien 2500 til at rotere med uret (fremad) og 500 til at rotere mod uret (baglæns).

For at robotten skulle kunne stoppe, definerede jeg variabler til venstre og højre hjul og gav dem begge værdien 1500, som svarer til servomotorernes neutrale position.

I setup() -funktionen initialiserede jeg alle relevante input og output med InitEmoro(). Derefter aktiverede jeg servomotorerne på SERVO_1- og SERVO_2-portene med emoroServo.attach(SERVO_0) og startede seriel kommunikation med Serial.begin(9600). Jeg aktiverede også input med pull-up på IR-sensorerne tilsluttet IO_1 og IO_2.

Jeg brugte kontrolstrukturer som if og else if til at teste IR-sensorernes signaler.

- I det første tilfælde opdager sensorerne en hvid overflade, og robotten kører fremad.
- I det andet tilfælde opdager de en sort overflade, og robotten stopper.
- Hvis kun højre sensor registrerer en sort overflade, bakker robotten og drejer til venstre.
- Hvis kun venstre sensor registrerer en sort overflade, bakker robotten og drejer til højre.

Billede af Arduino Robot Kode

```
Robot_Dikshy_ | Arduino 1.8.8
File Edit Sketch Tools Help

Robot_Dikshy_

static int LF =500; //left forward
// static keyword to create variables that are visible to only one function, created and initialized the first time a function is called
static int RF =2500;
static int LB =2500;
static int RB =500;
static int Lstop=1500;    // Servo values for each motor and each direction
static int Rstop=1500;    // Servo values for each motor and each direction

void setup(){
  InitEmoro();              // Initializes all available inputs and outputs on EMoRo 2560.
  EmoroServo.attach (SERVO_0);    // Initiates servo on SERVO_1 port.
  EmoroServo.attach (SERVO_1);    // Initiates servo on SERVO_2 port.

  Serial.begin(9600);           // initialize serial communication
  Serial.println("Example:IR+ servo");    // send example name to console

  pinMode(IO_0, INPUT_PULLUP);    // Initiates input mode with pullup - infrared sensor on IO_1 port.
  pinMode(IO_1, INPUT_PULLUP);    // Initiates input mode with pullup - infrared sensor on IO_2 port.
}

void loop (){
  if (digitalRead(IO_0) ==1 && digitalRead(IO_1)==1){                // Both sensros are on the white, go forward.
    EmoroServo.write(SERVO_0,LF);    // left servo forward
    EmoroServo.write(SERVO_1,RF);    // right servo forward
  }

  else if (digitalRead(IO_0) ==0 && digitalRead(IO_1)==0){            // Both sensors are on the black, stop.
    EmoroServo.write(SERVO_0,Lstop);    // left servo stop
    EmoroServo.write(SERVO_1,Rstop);    // right servo stop
  }

  else if (digitalRead(IO_0) ==0 && digitalRead(IO_1)==1){            // Only left sensor is on the black, turn right.
    EmoroServo.write(SERVO_0,LB);    // left servo backwards
    EmoroServo.write(SERVO_1,RF);    // right servo forwards
    delay(20);                    // delay 20 ms
  }

  else if (digitalRead(IO_1) ==0 && digitalRead(IO_0)==1){            // Only right sensor is on the black, turn left.
    EmoroServo.write(SERVO_0,LF);    // right servo forwards
    EmoroServo.write(SERVO_1,RB);    // left servo backwards
    delay(20);                    // delay 20 ms
  }
}

Done Saving.

Sketch uses 22296 bytes (8%) of program storage space. Maximum is 258048 bytes.
Global variables use 1471 bytes of dynamic memory.

28
```


Source Kode med kommentarer

Kode af traffic Lys fra arduino projekts bog:

```
int switchstate = 0;

void setup() {

  pinMode(3, OUTPUT);

  pinMode(4, OUTPUT);

  pinMode(5, OUTPUT);

  pinMode(2, INPUT);

}

void loop() {

  // put your main code here, to run repeatedly:

  switchstate = digitalRead(2);

  if (switchstate == LOW) {

    // the button is not pressed

    digitalWrite (3,HIGH); //green Led

    digitalWrite (4,LOW); //red Led

    digitalWrite (5,LOW); //red Led

  }

  else{ //the button is pressed

    digitalWrite (3,LOW);

    digitalWrite (4,LOW);

    digitalWrite (5,HIGH);

  }

  delay (250); // wait for a quarter second
```

```
//toggle the leds  
  
digitalWrite(4,LOW);  
  
digitalWrite(5, LOW);  
  
delay(250); // wait for a quarter second  
  
} // go back to the beginning of the loop
```

Kode af TrafikLys:

Den kode, jeg har brugt, til at vise, hvordan den virker, og hvad der er forskelligt mellem de to kode.

```
int switchstate = 0;  
  
int light = 0;  
  
int pinnum[] = {5,4,3};  
  
void setup() {  
  
    pinMode(3, OUTPUT);  
  
    pinMode(4, OUTPUT);  
  
    pinMode(5, OUTPUT);  
  
    pinMode(2, INPUT);  
  
}  
  
  
void loop() {  
  
    while (true) {  
  
        int presscontrol = 0;  
  
  
        for (int i = 0; 2 >= i; i++) {
```

```
digitalWrite(pinnum[i], HIGH);
```

```
delay(1000);
```

```
light = digitalRead(3);
```

```
switchstate = digitalRead(2);
```

```
digitalWrite(pinnum[i], LOW);
```

```
if (switchstate == HIGH && presscontrol == 0) {
```

```
    presscontrol = 1;
```

```
}
```

```
if (presscontrol == 1 && light == HIGH) {
```

```
    delay(1000);
```

```
    digitalWrite(pinnum[i], LOW);
```

```
    delay(1000);
```

```
    digitalWrite(5, HIGH);
```

```
    delay(3000);
```

```
    digitalWrite(5, LOW);
```

```
}
```

```
else {
```

```
}
```

```
}
```

```
}
```

```
}
```

Kode af Robot

```
/* Line following robot using 2 IR sensors*/
```

```
static int LF =500; //left forward
```

```
// static keyword to create variables that are visible to only one function, created and initialized the  
first time a function is called
```

```
static int RF =2500;
```

```
static int LB =2500;
```

```
static int RB =500;
```

```
static int Lstop=1500; // Servo values for each motor and each direction
```

```
static int Rstop=1500; // Servo values for each motor and each direction
```

```
void setup(){
```

```
    InitEmoro();           // Initializes all available inputs and outputs on EMoRo 2560.
```

```
    EmoroServo.attach (SERVO_0); // Initiates servo on SERVO_1 port.
```

```
EmoroServo.attach (SERVO_1);    // Initiates servo on SERVO_2 port.

Serial.begin(9600);              // initialize serial communication

Serial.println("Example:IR+ servo");  // send example name to console


pinMode(IO_0, INPUT_PULLUP);      // Initiates input mode with pullup - infrared sensor on
IO_1 port.

pinMode(IO_1, INPUT_PULLUP);      // Initiates input mode with pullup - infrared sensor on
IO_2 port.

}

void loop (){

  if (digitalRead(IO_0) ==1 && digitalRead(IO_1)==1){          // Both sensros are on the white,
go forward.

  EmoroServo.write(SERVO_0,LF);                                // left servo forward

  EmoroServo.write(SERVO_1,RF);                                // right servo forward

  }

  else if (digitalRead(IO_0) ==0 && digitalRead(IO_1)==0){      // Both sensors are on the black,
stop.

  EmoroServo.write(SERVO_0,Lstop);                             // left servo stop

  EmoroServo.write(SERVO_1,Rstop);                             // right servo stop
```

```
}

else if (digitalRead(IO_0) ==0 && digitalRead(IO_1)==1){      // Only left sensor is on the black,
turn right.

    EmoroServo.write(SERVO_0,LB);                          // left servo backwards

    EmoroServo.write(SERVO_1,RF);                          // right servo forwards

    delay(20);                                              // delay 20 ms

}

else if (digitalRead(IO_1) ==0 && digitalRead(IO_0)==1){      // Only right sensor is on the black,
turn left.

    EmoroServo.write(SERVO_0,LF);                          // right servo forwards

    EmoroServo.write(SERVO_1,RB);                          // left servo backwards

    delay(20);                                              // delay 20 ms

}

}
```

Konklusion:

Jeg har lært nogle grundlæggende ting, som jeg har brug for at kunne for at løse Arduino-projektet. For eksempel hvad en servo er, og hvordan servo-værdierne for hver motor og retning kan initialiseres (højre/venstre, fremad, bagud, forsinkelse og stop). Jeg har også lært at initialisere alle tilgængelige input og output, samt forskellen mellem input og output.