



# TTT4255 Elektronisk systemdesign, grunnkurs

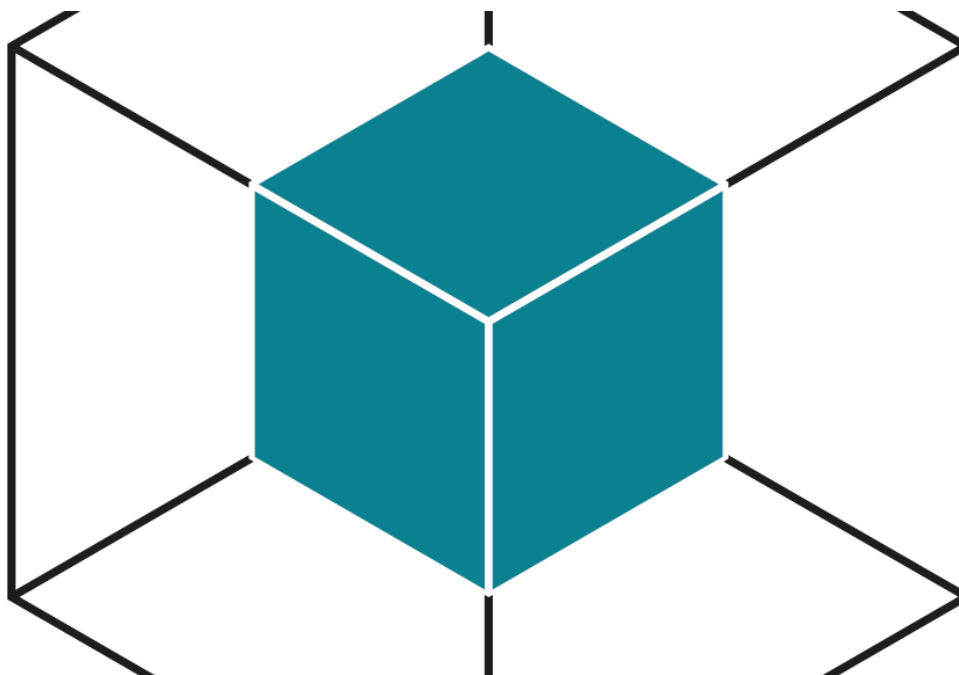
## P1: Whack a LED

Elektronisk systemdesign og innovasjon

---

Ida Bjørnevik, Sven Amberg, Amalie 29.06.2023  
Fridfeldt Hauge og Peter Magerøy

---



### Innhold

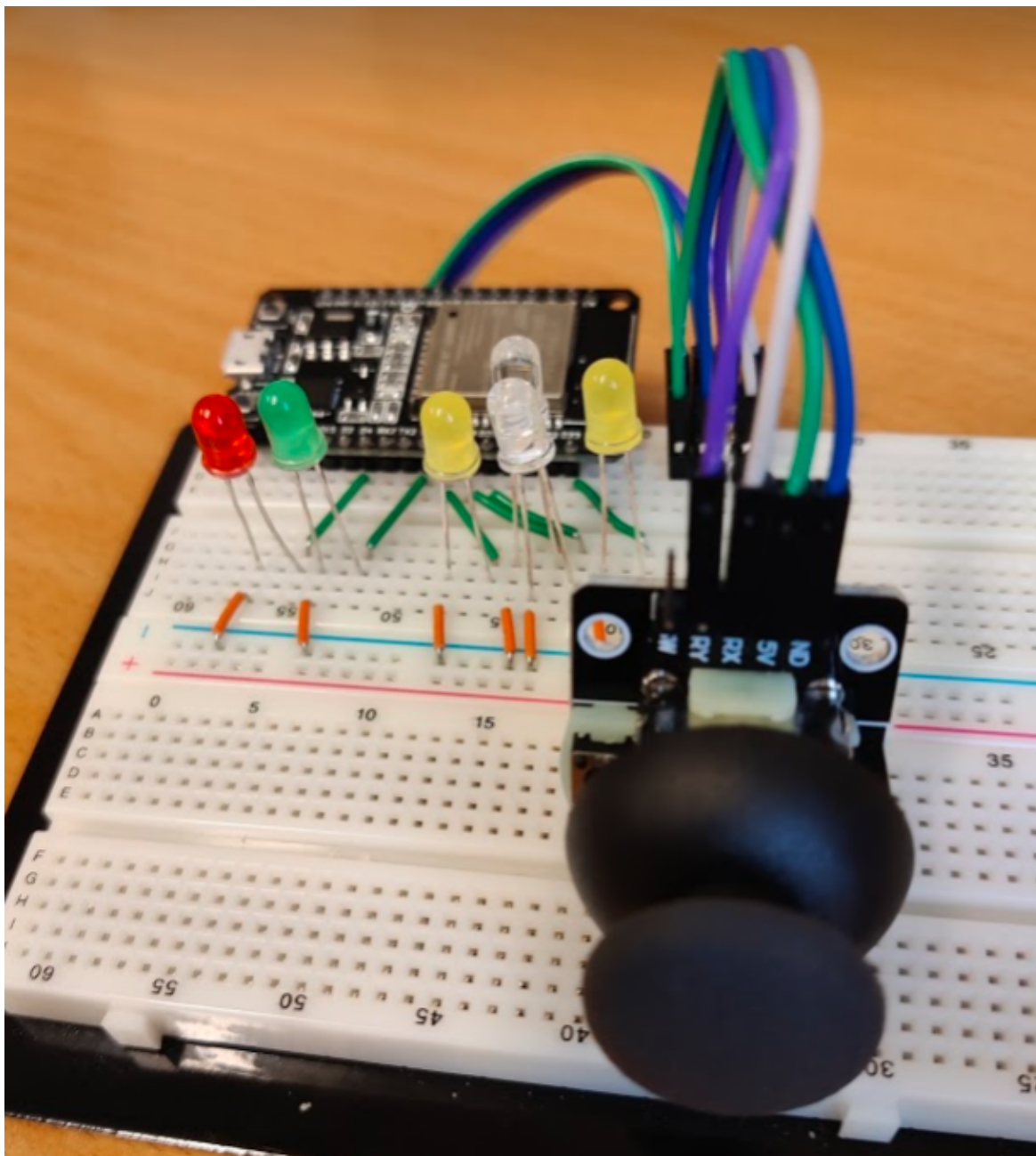
<b>Introduksjon</b>	<b>2</b>
<b>Prosjektet: Kort forklart</b>	<b>3</b>
Utstyrliste . . . . .	3
Steg 1: LED . . . . .	3
Steg 2: Joystick . . . . .	3
Steg 3: Programmering . . . . .	4
<b>Prosjektet: Mer grundig forklaring</b>	<b>6</b>
Steg 4: Programmering . . . . .	6

---

## Introduksjon

Passar for deg med som har kunnskap frå **A1-LED** og **S4-Joystick**.

I dette prosjektet skal vi lage vår egen versjon av spillet "Whack a mole", men i stedet for å slå en muldvarp skal vi "slå" en lysdiode med joystick.



Figur 1: Whack a LED!

## Prosjektet: Kort forklart

### Utstysrliste

- ESP32
- 1 grønn LED
- 1 rød LED
- 4 LED-er i valgfrie farger
- Joystick

### Steg 1: LED

Lysdiodene skal kobles til jord og seks valgfrie GPIO, det vil si at det lange beinet til lysdioden skal kobles til GPIO og det korte til jord. Se grunnleggende modul A1 for tips til å koble LED. Se plassering av lysdioder i figur 2.



**Figur 2:** Plassering av lysdioder.

### Steg 2: Joystick

Koble opp joysticken slik at:

- Xout - GPIO
- Yout - GPIO
- GND - GND
- Vcc - 5 V

### Steg 3: Programmering

Skriv inn egen kode der det er markert med `/*` og `*/`. Husk å legge inn verdi på Xpin og Ypin, dette skal samsvare med de GPIO **Xout** og **Yout** som joysticken er koblet til.

```
//Pin joysticken er koblet til:
const int X_pin =    ;
const int Y_pin =    ;

//Globale variabler:
int wins = 0;
int highscore = 0;

//Lista er nullindeksert og skal følge nummerering på LEDs som
//er avbildet.
int ledPins[] = { , , , , , };
int pinCount = 6; //Antall pins kopla til LEDs.
```

**Figur 3:** Variabeldeklarasjoner.

```
void setup() {
    Serial.begin(115200);
    /*
     * Sett pinMode til X_pin og Y_pin.
     */

    /*
     * Sett pinMode til LEDs.
     */
}
```

**Figur 4:** Setup-funksjonen i programmet.

```

void loop(){
    int randNum = random(4); //Genererer tilfeldig tal mellom 0
        og 3.
    delay(1500);
    /*
        * Få en tilfeldig LED til å lyse.
        */
    delay(); //Tida du har til å reagere.
    //Les av joysticken
    int x = analogRead(X_pin);
    int y = analogRead(Y_pin);

    int s;
    /*
        * Gi s verdi etter kor joysticken er, sjå figur.
        * Hint: Bruk fire if-setningar.
        */
    /*
        * Skru av LEDen.
        */

    if(s == randNum){ //Ser om joysticken stod rett.
        wins++;
        /*
            * Skru av og på grøn LED.
            */
    }

    else{
        if(wins > highscore){
            highscore = wins;
            /*
                * Gratuler spelaren med sigeren i seriellovervåkeren.
                */
        }
        /*
            * Skru av og på raud LED.
            */
        /*
            * Print resultat og highscore til seriellovervåker.
            */

        wins = 0;
    }
    delay(500);
}

```

**Figur 5:** Loop-funksjonen i programmet, det er her spillogikken blir implementert.

Bruk gjerne tidligere moduler som hjelpemiddel for å skrive koden.

## Prosjektet: Mer grundig forklaring

### Steg 4: Programmering

Åpne Arduino IDE og skriv av koden i figurene under. Først definerer vi pins til X og Y fra joysticken, dette skal samsvare med den fysiske kretsen, figur 6. Du kan fritt bruke andre GPIO-pins enn det som er vist i eksemplene, husk da å bruke pins for input/output (vist i intro modul x).

```
//Pin joysticken er kopla til:
const int X_pin = //GPIO;
const int Y_pin = //GPIO;

//Globale variabler:
int wins = 0;
int highscore = 0;

int ledPins[] = { //GPIO for LEDs}; //Liste med pins der LED er
kopla til.
int pinCount = 6; //Antall pins til LED.
```

**Figur 6:** Variabeldeklarasjoner.

I setup-funksjonen (figur 7) setter vi opp hva som er formålet med pins som vi har definert. I tillegg startar vi seriellovervåkeren med baudrate 115200. Det er brukt ei for-løkke for å sette alle LED-pins til OUTPUT.

```
void setup() {
    Serial.begin(115200);
    pinMode(X_pin, INPUT);
    pinMode(Y_pin, INPUT);

    for (int i = 0; i < pinCount; i++) {
        pinMode(ledPins[i], OUTPUT); //Går gjennom alle LEDpin.
    }
}
```

**Figur 7:** Setup-funksjonen i programmet.

I loop-funksjonen er spillokikken implementert. Her blir det først generert et tilfeldig tall, dette blir brukt for å få en tilfeldig LED til å lyse. Deretter blir det lest hvilken posisjon joy-

stikken har, dette blir lagret i variabelen `s`. Deretter slutter LED-en å lyse. Dersom posisjonen peker på rett LED blinker den grønne LED-en og variabelen `wins` blir oppdatert til den nye scoren.

Siste del av programmet blir kjørt dersom en bommer på LED-en. Her blir highscore oppdatert dersom en får ny rekord. I tillegg lyser den røde LED-en for å indikere at en har bommet. Resultatet blir for hver runde vist i seriellmonitoren.

```

void loop(){
    int randNum = random(4); //Genererer tilfeldig tal.
    delay(1500);
    digitalWrite(ledPins[randNum], HIGH); //Skrur på tilfeldig
    LED
    delay(500); //Tida du har til å reagere.

    //Les av joysticken
    int x = analogRead(X_pin);
    Serial.println(x);
    int y = analogRead(Y_pin);
    Serial.println(y);
    int s = 4;
    if(x < 2000){s = 3;} //Kort form av if-setning, sjekkar kva
        retning joysticken peikar.
    if(x > 3000){s = 0;}
    if(y < 2000){s = 1;}
    if(y > 3000){s = 2;}
    digitalWrite(ledPins[randNum], LOW); //Slår av LEDen.
    if(s == randNum){ //Ser om joysticken stod rett.
        wins++;
        //Skrur av/på den grønne LEDen.
        digitalWrite(ledPins[5], HIGH);
        delay(500);
        digitalWrite(ledPins[5], LOW);
    }

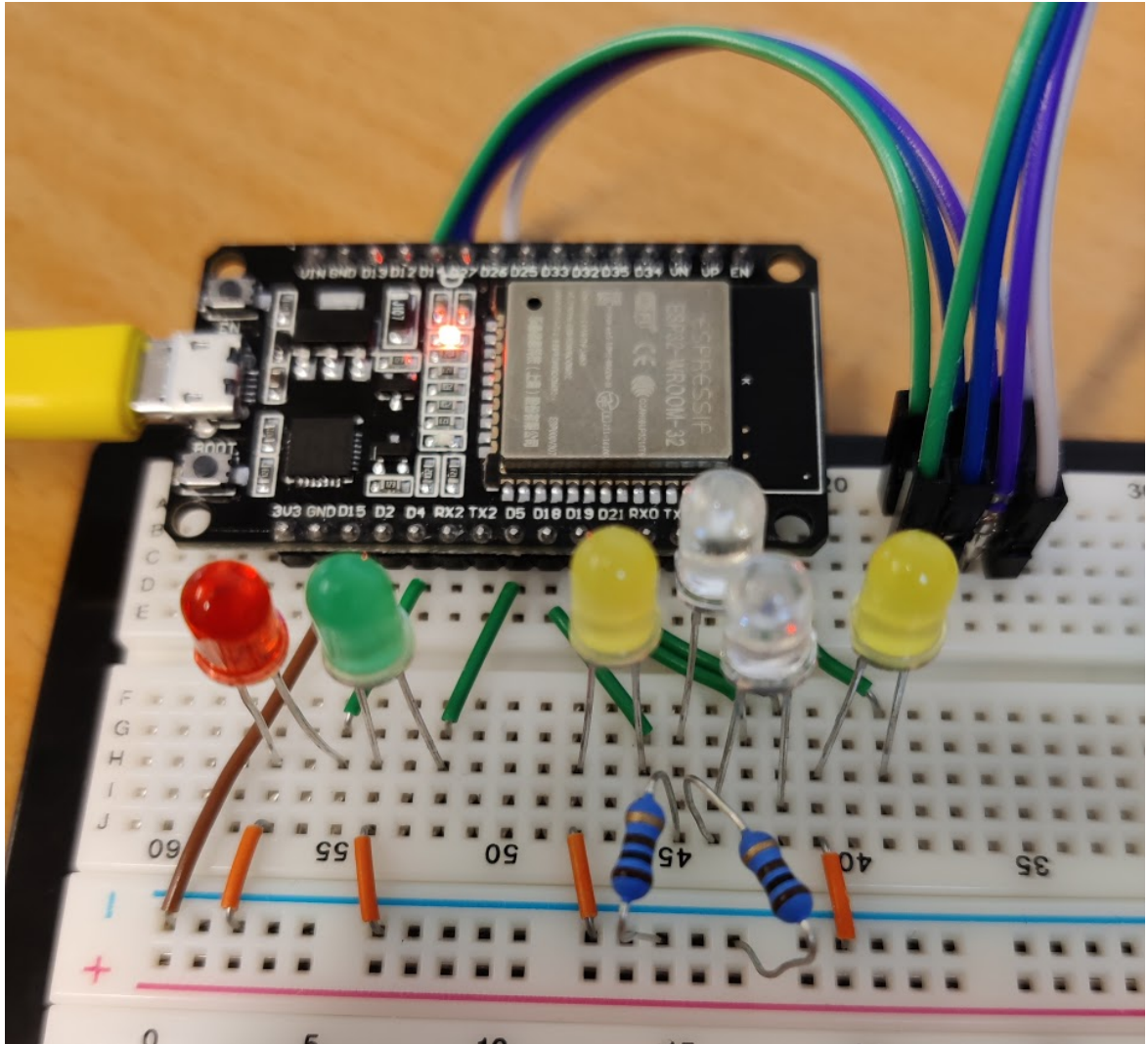
    else{
        if(wins > highscore){
            highscore = wins;
            Serial.println("Gratulerer du har satt ny rekord");
        }
        digitalWrite(ledPins[4], HIGH);
        delay(500);
        digitalWrite(ledPins[4], LOW);
        //Skriv ut resultat.
        Serial.print("Correct: ");
        Serial.print(wins);
        Serial.print(" Highscore: ");
        Serial.println(highscore);
        wins = 0;
    }
    delay(500);
}

```

**Figur 8:** Loop-funksjonen i programmet, det er her spillogikken blir implementert.



Et ferdig oppsett med en annen mikrokontroller er vist i figur 9. Spør hverandre og lærings-assistentene dersom dere står fast eller lurere på noe.



9

#### Ekstra funksjoner (frivillig):

- Legg til funksjonalitet slik at når en får ny high score pauser spillet i 2 s og alle lysdiodene lyser.
- Legg til buzzer (modul A6) med lydeffekter, for eksempel når du får en ny high-score eller når en bommer.
- Legg til skjerm (modul A4 eller modul A5) som viser scoren.
- Bruk modul S5 og bytt ut joystick med touch sensorer.
- Vær kreativ og finn på noe gøy!