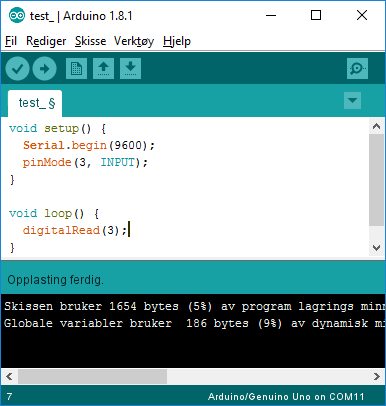
## **Knapp – Forklaring**

Nedenfor ser du to forskjellige representasjoner av oppkobling av en knapp. Til venstre har du det du kaller en skjemategning og til høyre, en illustrasjon som er lik en virkelig oppkobling. Begge disse er ekvivalente, dvs. de er like. Det er ofte lurt å bruke illustrasjonen til å få et overblikk og skjemategningen til å verifisere koblingen.

|  |  |
| --- | --- |
|  |  |

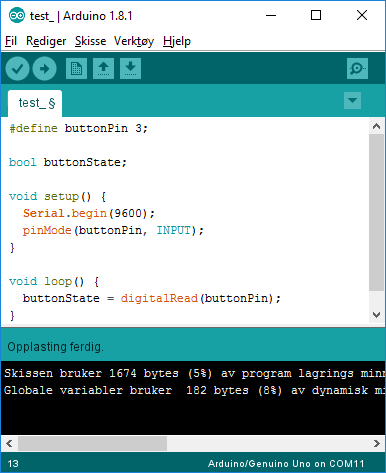
For å «lese» av knappetrykket bruker du en funksjon som heter digitalRead(pin) den leser av en digital verdi, dvs. av eller på. Av eller på er representert som 0 eller 1, false eller true, low eller HIGH. Pin er ett tall mellom 1 og 13, som representerer de nummererte plassene på Arduinoen. digitalRead funksjonen returnerer HIGH hvis det er spenning på pinnen og LOW hvis det ikke er spenning på pinnen. Ved å trykke på knappen setter vi spenning på pinnen og funksjonen returnerer HIGH.

Til høyre ser du hvordan du leser av verdien til en pinne. Det første du bør legge merke til er at vi først må fortelle Arduinoen at pinne nummer 3 er en INPUT dvs. at det er en inngang.

## **Knapp – Eksempel 1**

I dette eksempelet skal vi gå gjennom hvordan du leser av en knapp og bruker dette til å utføre en handling. I forklaringen fikk dere se hvordan du leser av en knapp, men mikrokontrollere er ikke «smarte». Hvis vi sier «les av en knapp» så leser den knappen, ikke noe mer. Vi må da enten bruke avlesningen i det den blir tatt eller lagre resultatet av avlesningen i en variabel.

Dette eksempelet er representativt av hvordan du bruker en knapp. Det første som er gjort er å definere buttonPin 3, dette betyr at hver gang vi skriver buttonPin så bytter IDE

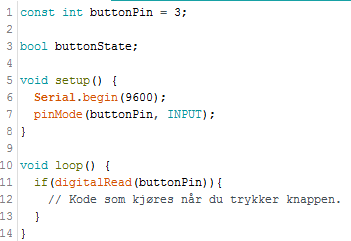
Notis:

|  |
| --- |
| #define buttonPin 3  er ekvivalent med  const int buttonPin = 3; |

Legg merke til at i dette eksempelet tar vi avlesning av knappen og lagrer dette i variabelen buttonState.

## **Knapp – Eksempel 2**

Dette eksempelet viser hvordan du kan bruke en digital avlesning av en knapp uten å lagre noe til en variabel. Dette er ofte brukt i sammenheng med if-statements

Til høyre kan du se at digitalRead(buttonPin) er satt som argument til en if-løkke. digitalRead(buttonPin) funksjonen returner enten «1» eller «0» basert på om det er spenning eller ikke på pinnen.

If-løkken kjøres hvis det står if(1), den kjøres ikke når det står if(0).

## **Toggling – Eksempel 1**

Skriv av og test denne koden nedenfor. Dette er en kode som toggler en boolsk variabel. Denne variabelen kan brukes til å styre andre funksjoner, leds, if-statements osv. OBS: Du må også koble opp en knapp på pin 2 og led på pin 11 (eller det du selv vil).

#define buttonPin 2

#define redLed 11

bool buttonWasReleased = false;

bool toggleValue = false;

void setup() {

**Serial**.begin(9600);

 pinMode(buttonPin, INPUT);

 pinMode(redLed, OUTPUT);

}

void loop() {

 if (digitalRead(buttonPin) && buttonWasReleased) {

   buttonWasReleased = false;

   toggleValue = !toggleValue;

 }

 if (!digitalRead(buttonPin)) {

   buttonWasReleased = true;

 }

 digitalWrite(redLed, toggleValue);

**Serial**.println(toggleValue);

}

Her vil du se at du kun kan endre verdien til toggleValue, hvis du først slipper opp knappen.