

# Tidtagarur

## Mål med Projekt

Målet med följande projekt var att skapa ett tidtagarur som kan ta varv och se vad tiden för varje varv skulle vara, genom att skrolla upp och ner i varven, likt en tidtagarur som man har på en mobiltelefon. Man ska även kunna pausa tiden och låta den fortsätta om man så vill. Tidtagaruret går upp till en timme, där det kommer att stanna eftersom man endast går på minuter, sekunder och millisekunder med två decimaler. För att kolla vilket varv man är på går det att se tack vare led-lamporna.

## Lösning

För detta projekt tog jag och använde endast labbkittet (Chipkit Uno32 med Basic I/O Shield). OLED skärmen använder för att visa allt data på skärmen, högst upp tiden och resten av skärmen tiderna för varje resultat av de förra varven. Knapparna används till för att pausa/fortsätta, ta nytt varv och skrolla mellan varven. I början hade jag sista knappen som en knapp för att starta om men tog bort denna eftersom man hade så få knappar att välja mellan och det finns en "reset" knapp på chipkittet redan.

Knapparna skulle även ta och reagera instant. I detta projekt använder jag mig utav "polling" för att använda knapparna, där det finns en variabel som blir 1 när knappen är nedtryckt och blir sedan 0 när den släpps. Jag använde mig av "polling" istället för "interrupts" då enbart switcharna är kopplade till "Extended interrupt" schemat, då jag ville använda knapparna för detta projekt.

Under labbarna har vi lärt oss att man kan skriva ut bestämda strängar och valfria heltal på skärmen med hjälp av det vi har fått. Men för att kombinera valfri text med valfritt heltal har jag använt mig av C bibliotekets inbyggda funktion sprintf, vilket är en funktion som kombinerar en char array med integers, likt en string kan göras i exempelvis Java. Jag ville göra det i detta fall då jag ville skriva ut på skärmen exempelvis följande: "LAP 1: 12:34,56". När det blir ett nytt varv kommer led lampan att skifta ett steg till vänster för att påpeka vilket varv det är.

## Verifikation

Verifiering har gjorts för att undvika olika buggar som kan förekomma, samt därmed löst dessa. Exempelvis har knapparna något som heter "bouncing", vilket betyder att en när man släpper en knapp kommer den tro att den trycks ner igen i några mikrosekunder,

går att se att detta är fallet genom att kolla ledlamporna på inputten. För att lösa detta fick man ta och implementera en pytteliten delay som man fick gratis när man gjorde första laborationen. Den är inte exakt men det är väldigt liten att den knappt märks av utav användaren och tillräckligt liten för att just motverka "bouncing". Men eftersom "delay" funktionen fryser programmet använder vi denna typ av funktion endast på nytt varv och paus/fortsätt knappen, så programmet kommer köras som vanligt när man skrollar upp och ner. Om bouncing sker i skrollingen är det inte stor påverkan i jämförelse med om det skulle ske när man exempelvis tar ett nytt varv.

### **Bidrag**

Eftersom jag har arbetat ensam hela projektet har jag tagit och skrivit all kod i detta projekt om man bortser från de saker som inkluderades från laboration 3.

### **Reflektioner**

För detta projekt har jag lärt mig ganska mycket syftet med att använda exempelvis BITWISE operationer för att lösa olika problem. Eftersom jag endast har använt mig av högnivåspråk som Java och Python fick det mig att se logiken bakom C programmering och även hur man kan använda sig av inbyggda system, vilket även är en kurs vi kommer att ha i årskurs 2 och dessa förkunskaper kommer att underlätta.