Anders Mikkelsen

Innlevering 2

Oppgave:

Lage en vekkerklokke, som skal vise korrekt tid og kunne spille av alarm.

Avhengigheter:

- Wire, SPI, SoftwareSerial
- RTClib https://github.com/adafruit/RTClib
- Adafruit GFX https://github.com/adafruit/Adafruit-GFX-Library
- Adafruit ST7735 https://github.com/adafruit/Adafruit-ST7735-Library
- IRlib https://github.com/cyborg5/IRLib (Må settes til Timer 1)

Dokumenter:

- Dette
- Fritz fil
- Fritz BB Screen Capture som .png
- Videodokumentasjon (Beklager kvaliteten, rævva webcam og vanskelig å filme skjermen med det): https://www.youtube.com/watch?v=wyYdxGGpLp4&feature=youtu.be&hd=1

Design:

Ambisjonen min når jeg startet var å lage en vekkerklokke med interaktiv skjerm. Altså at man skulle kunne rotere den som man ville og ha skjermen til å følge etter og alltid være horisontal ift. brukeren. I tillegg ville jeg ha en graderende farge basert på tid på døgnet, hvilemodus og tidsynkronisering via internet. Jeg hadde også en ambisjon om å hente ut SSID og Pass fra SDkortet.

Implementasjon:

Jeg har forsøkt å skrive kode som tar lite plass og dytter det meste av de store variablene (arrays osv) inn i PROGMEM, slik at jeg får det ut av SRAM. Jeg synes jeg har endt opp med en ganske sunn buffer på SRAMen. Den rapporterte bruker for globals er 50% (1,026 b). Dette burde være mer enn godt nok da jeg har relativt korte functions og bruker så mye pass-by-reference og pointers som mulig.

Jeg har i denne innleveringen slitt litt med timers da jeg prøver å gjøre mye på alle pins. Jeg måtte derfor finne en løsning som funker. Derfor kan pinfordelingen ved første øyekast se noe merkelig ut. Jeg har koblet opp sdkortet fordi jeg hadde ambisjoner om å bruke det men

jeg tror jeg er tom for plass på sketchen. Valget stod mellom det og IR / Wifi. Da synes jeg det var ett enkelt valg. Sketchen ligger nå på 100% (32,256 b) progmem med generisk ssid og pass. (Kan hende jeg la på noen tegn fordi det var litt gøy å treffe grensa.)

Etterhvert gikk jeg tom for pins. Dvs, jeg kunne koblet ut sden og hatt flere pins, men har en ambisjon om å refaktorere og komprimere til ett nivå der jeg kan inkludere dette. Derfor har jeg benyttet meg av pin 0 til input for IR. Dette var lærerikt! For det første tok det en del banning til før jeg forsto at failed upload var fordi jeg hadde ledning koblet i pin 0. Så det har vært litt irriterende å måtte ta ut den for å uploade men det har vært bagatellmessig. IRlib er satt opp med Timer 1 i stedet for Timer 2 (gikk inn i header og endret, dette må du også gjøre lokalt) men det gjelder kun for IRsending. Grunnet til at denne endringen er nødvendig er at tone() kjører på timer 2. Jeg vil også anta at å forsøke det på dette oppsettet ikke vil fungere da tften prater på pinsene til Timer 1.

Når jeg satte i gang med IRlib slet jeg en del med å få ting til å reagere som jeg trodde det ville. Det så ikke ut som arduinoen reagerte når jeg sendte commands og jeg begynte etter hvert å mistenke fjernkontrollen. Jeg bestemte meg derfor for å teste fjernkontrollene jeg hadde i sofaen for en som var NEC. Yamaha Fjernkontrollen til receivern min viste seg å være på NEC protokoll og jeg brukte den i stedet. Da mottok arduinoen klare og fine IR signaler og alt fungerte som smurt. Dette var også en bra læringsmulighet!

Jeg opplevde plutselig en dag mot slutten av innleveringsperioden at ESP8266 modulen min hang seg og ikke ville kommunisere med meg mer. Den koblet fortsatt opp på siste wifinett jeg hadde koblet den med men nektet å kommunisere. Etter mail til deg stakk jeg og kjøpte en ny. Den var «brand spanking new» også på sdk og at commands og den fungerer ypperlig. ESPen skriver dessverre ut en del garbage. Jeg vet ikke om dette er en baudsetting eller noe i den duren. Vil gjerne finne ut av det. Jeg kjører på baud 57600 nå. Derfor vil tiden kunne være upresis. Tidshentingen gjør at klokken automatisk håndterer Daylight Savings Time. (Norsk område).

I starten brukte jeg delay() for å gjøre kontinuerlige ting, som f.eks. dimmingen men endte med å fjerne alt det og heller kjøre ett timersystem så jeg slipper å forholde meg til delays her og der. Jeg synes det er en bedre løsning.

Jeg synes jeg har løst oppdraget og er happy med resultatet. Nærmere beskrivelse av bruk er i videoen.

Bugs:

- Jeg hadde en stund en bug som frøs hele arduinoen. Det ligger i pulseIn functionen som jeg bruker på ultralydkomponenten. Etter å ha saumfart google litt for info så jeg at mange andre hadde hatt issues med dette. Dette forsvant imidlertid etter hvert så jeg vet ikke hva som var issuet.

Fritzing:

Ingen overraskelser her, det litt «snodige» pinvalget på enkelte komponenter har jeg beskrevet lenger opp. Noen av kabelfargene vil variere fra fritz til reel utgave og det er fordi jeg ikke hadde flere av fargene jeg ønsket. Fargevalgene på fritzen er å regne for «korrekt».

Eksterne ressurser:

Generelt: Arduino.cc

AT Commands og ESP debug: https://nurdspace.nl/ESP8266