Alarm Clock

Nume: Sandru Mihaela-Ilinca
Grupa: 335CA

Introducere

Tema proiectului reprezinta un ceas digital, care are posibilitatea de a-i seta un numar maxim de 4 alarme. Are optiunea de a-i configura ceasul si luminozitatea display-ului. In plus, ceasul afiseaza temperatura si umiditatea din camera, iar pentru anumite valori critice ale acestora se declanseaza pornirea unui umidificator alimentat prin usb-c.

Ideea a luat nastere de la niste cuburi de iluminat LED (Frekvens) cumparate de la Ikea. Animatiile facute pe baza muzicii nu erau pe placul meu. Prin urmare, am desfacut cutia si am descoperit ca fiecare led este comandat individual prin shift-registere.

Ceasul cu alarma este util pentru orice persoana. Un telefon din zilele noastre rezolva aceasta problema usor si rapid, dar ceasul meu ofera facilitati suplimentare si are un design deosebit.

Descriere generală

In cadrul proiectului, folosesc 3 placi A-Star ATmega328PB, dispuse intr-o arhitectura de tipul Master-Slave-Slave sau CPU-GPU-GPU, intrucat cele 2 placi Slave se ocupa de partea de “desenare” pe matricea de led-uri, iar Masterul le comanda. Masterul comunica cu Slave-urile prin protocolul de comunicatie UART, iar Slave-urile comunica cu shift-registerele prin protocolul SPI simplex.

Pentru ca este vorba de un ceas, folosesc un modul extern RTC (Real Time Clock) pentru a tine evidenta timpului. Acest modul comunica cu Masterul prin protocolul I2C. Alarmele se configureaza prin intermediul butoanelor de pe spatele cuburilor Frekvens. La momentul setat, se va declansa o alarma prin difuzor. De asemenea, Masterul primeste informatii de la senzorul de temperatura si umiditate, iar la anumite valori critice ale acestora va porni umidificatorul prin intermediul mosfetului.

Block diagram

Hardware Design

Lista de componente:

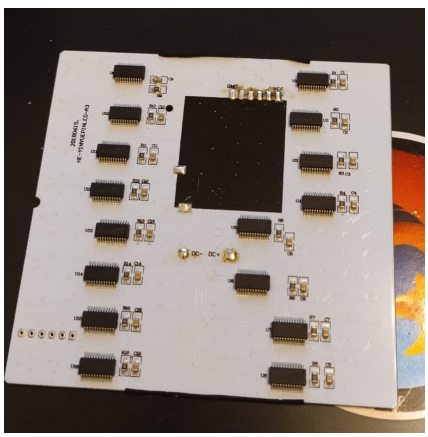
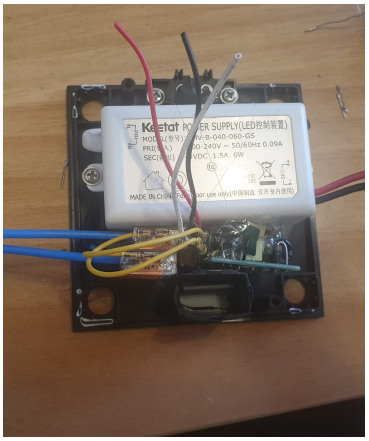
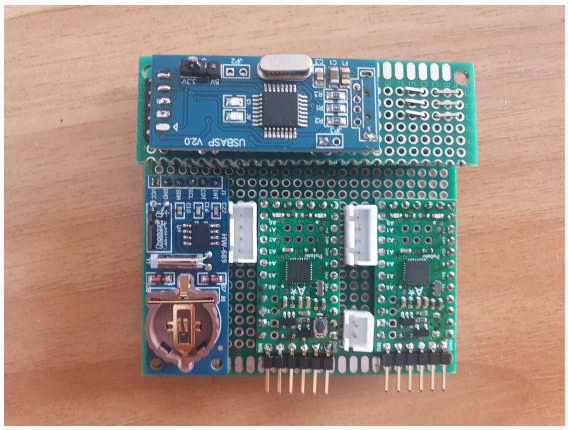
- 2 cuburi Frekvens (32 shift-registere SCT2024 + 256 LED-uri + 4 butoane)
- 3 placi A-Star ATmega328PB
- amplificator audio PAM8403
- difuzor
- senzor de temperatura si umiditate DHT11
- modul de PCF8563 RTC (Real Time Clock)
- 2 USB-C
- mosfet AO3400
- USBASP AVR Programmer
- 2 AC-DC Power Supply (una de 3.3v si una de 5V)

Electrical Schematic

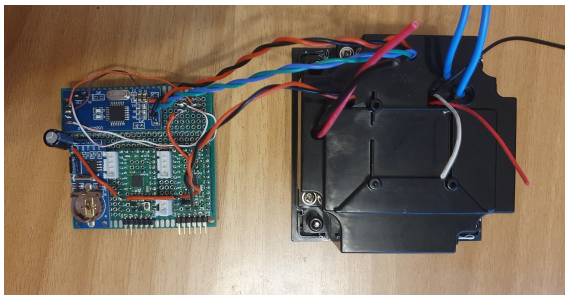
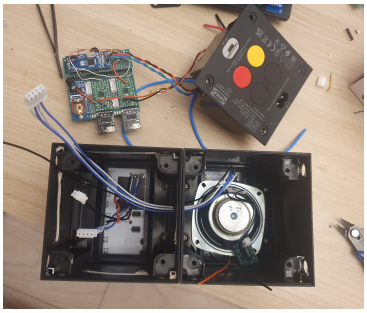
electrical_schematic.pdf

Hardware Journey

De pe placile cu leduri si shift-registere am scos chip-ul care le controla. Am pus cele 2 surse izolate de 3.3V si 5V si USB-C-urile in capacele cuburilor Frekvens, izoland astfel traseele de putere de cele de semnal. Pe placuta de prototipare am lipit cele 3 placi A-Star ATmega328PB, USBASP-ul, modulul RTC si firele dintre ele si am plasat-o in cutia numarul 1 (stanga).

In cutia numarul 2 (dreapta) am adaugat difuzorul cu amplificatorul audio PAM8403 si mosfetul AO3400. Si am scos senzorul de temperatura si umiditate in afara cutiei.


Software Design

Flow Chart

Codul se bazeaza pe un automat de stari, conform diagramei de mai jos.

Automatul porneste din starea Clock in care afiseaza pe display ora curenta primita de la modulul RTC. La o apasare lunga a butonului Select (de aici inainte vom trata acest eveniment ca buton de Return) se va afisa pe ecran intensitatea luminoasa a ecranului, care poate fi modificata prin butoanele de Up si Down.

Prin apasarea butonului Select in starea Clock se ajunge in starea Menu, in care se va afisa pe ecran optiunile pe care le are utilizatorul: setarea a patru alarme, configurarea ceasului si a valorii critice la care sa porneasca umidificatorul (mosfetul), afisarea temperaturii si umiditatii. Modificarea valorilor se face prin intermediul butoanelor Up, Down si Select (acolo unde este cazul).

Revenirea dintr-o stare la o stare anterioara se face prin butonul Return (valabil pentru orice stare).

Code Structure

Codul este impartit in doua mari categorii de fisiere: cele destinate componentei master (master.ino, button.h, settings.h), respectiv cele pentru slave-uri (slave.ino, standard_matrices_draw.h).

Slave

In fisierul standard_matrices_draw.h, se gasesc matricile pe care le foloseste slave-ul ca sa deseneze pe display, salvate in PROGMEM (memoria flash).

In fisierul slave.ino este implementata logica pentru componenta slave. Sunt initializate o interfata UART Software (comunicarea cu masterul), o interfata seriala de tip hardware pentru debugging si pinii pentru shiftarea bitilor. Slave-ul asteapta inputuri de la master si pe baza comenzilor primite deseneaza in frame buffer imaginea ceruta si shifteaza bitii la output. Inputurile sunt de forma *<nr_slave><luminosity><command>*, unde *nr_slave* reprezinta numarul slave-ului pentru care este

destinat mesajul, iar *luminosity* se foloseste la realizarea PWM-ului pentru reglarea intensitatii led-urilor.

Master

Masterul comunica cu slave-urile prin Software UART. La fiecare trecere prin loop, masterul:

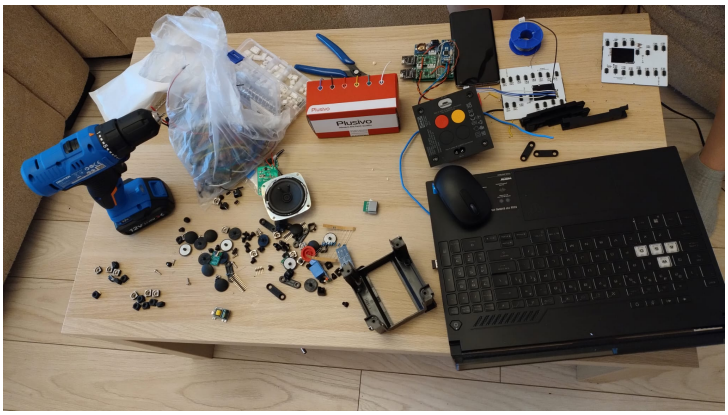
- citeste ceasul de la modulul RTC si trimite comanda la slave o data pe minut, atunci cand e in starea CLOCK;
- verifica starea alarmelor (porneste difuzorul timp de un minut la momentul configurat de alarme);
- verifica umiditatea din aer (daca este peste o valoare critica setata,va porni umidificatorul);
- schimba starea automatului in functie de actiunile butoanelor.

Alarmele, valoarea critica pentru umiditate si luminozitatea sunt salvate persistent in EEPROM.

Rezultate Obţinute

Am pornit de aici!

Am trecut pe aici 😎



Si am ajuns la:

Concluzii

Acest proiect a fost complex atat din punct de vedere hardware, cat si software.

Am invatat despre interfatarea software-ului cu hardware-ul, despre stocarea datelor in PROGMEM si EEPROM.

Desigur, am intampinat multiple probleme:

- de cateva ori, am facut lipituri reci, asa ca s-au desprins din loc;
- cutiile sunt facute sa fie asamblate/dezasamblate foarte greu;
- declararea unei variabile locale intr-o ramura de case afecteaza functionarea switch-ului (compilatorul presupune ca varful stivei ramane la fel pe tot parcursul switch-ului);
- am avut nevoie de un delay mare de debouncing, deoarece butoanele sunt springy.

Download

alarm_clock.zip

Jurnal

- 7 aprilie: documentare asupra unei teme de proiect
- 12 aprilie: alegerea temei de proiect
- 18 aprilie: comandarea componentelor de la magazinul Optimus Digital
- 27 aprilie: realizarea documentatiei initiale (introducere, descriere generala, schema bloc, schema electrica)
- 24 aprilie - 16 mai: realizarea componentei hardware
- 17 mai - 28 mai: realizarea componentei software
- 29 mai: definitivarea documentatiei

Bibliografie/Resurse

Datasheet

Datasheet ATmega328PB: https://ww1.microchip.com/downloads/en/DeviceDoc/40001906A.pdf
Datasheet RTC PCF8563: https://www.nxp.com/docs/en/data-sheet/PCF8563.pdf
Datasheet DHT11 sensor: https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf
Datasheet SCT2024 shift register: http://www.starchips.com.tw/pdf/datasheet/SCT2024V01_03.pdf

Another

https://reference.arduino.cc/reference/cs/language/functions/analog-io/analogwrite/
https://reference.arduino.cc/reference/en/language/variables/utilities/progmem/
https://docs.arduino.cc/learn/built-in-libraries/eeprom

Export to PDF