

Technika Mikroprocesorowa

Laboratorium

Harmonogram.

10 zajęć po 4,5 godziny

1. BHP, zasady.
Omówienie realizowanych tematów, stanowisk, sprzętu, środowiska. Wprowadzenie do programowania: pętla setup, loop, instrukcje if, for, while, switch, funkcje delay(). Zmontowanie układu z LEDem i mruganie nim na różne sposoby.
2. Obsługa UART.
Komunikacja z komputerem. Obsługa przez terminal. Wykresy w Kreślarce. Komunikacja pomiędzy uC.
3. Obsługa GPIO, analog. Przerwania zewnętrzne. Input Capture.
Moduł RTC (DS1302), enkoder kwadraturowy (IDUINO encoder).
Klawiatura gpio/analog (DFROBOT DFR0792), czujnik temp. ntc 10k, czujnik akcelerometr analogowy (ADXL335).
4. Obsługa I2C.
Czujnik IMU (DFROBOT SEN0252), enkoder (AS5600).
5. Obsługa SPI.
Wyświetlacz graficzny LCD (LCD TFT 1,8" 128x160px ST7735S). Moduł karty SD na SPI.
6. Czujniki różne.
Pulsoksymetr (DFROBOT SEN0518 - MAX30102), EMG (DFROBOT SEN0240), czujnik CO2 (SGP30), czujnik propanu (MQ-2) (+spray).
7. Obsługa PWM i sterowanie serwem.
Serwo modelarskie. Serwo Dynamixel XL330-M288-T.
8. Projekt własnego układu sterowania (minimum: uC, przycisk, czujnik, wysyłanie danych na komputer) - zamiast własnego, zrobić im regulator PID na grzałce i wiatraczku.
9. Kontynuacja projektu własnego.
10. Kontynuacja projektu, zaprezentowanie go. Wystawianie ocen.

Opis.

Sprzęt:

- 8 komputerów PC, internet
- 8 pudełek z zestawami

Lab1.

PRE:

Komputer z Windows, internet (lub instalki już na dysku), pudełko z uC i czujnikami

POST:

Zainstalowane środowisko Arduino IDE z bibliotekami do ESP32

Uruchomiony LED: on, off, toggle, mruganie z różnymi czasami włącz i wyłącz.

Przećwiczone operatory, instrukcje i podstawowe funkcje.

OPIS:

- Wstęp, bhp, warunki zaliczenia. Omówienie realizowanych tematów, stanowisk, sprzętu, środowiska. - 1h
- Instalacja środowiska, sterowników, pierwsze uruchomienie uC. - 1h
- Zbudowanie układu z LEDem. Przypomnienie symboli na schematach elektronicznych. - 1h
- Ćwiczenia z programowania: -1,5h
 - dodać `serial.begin(115200)`; żeby wyświetlać/czytać na comie informacje, ograniczyć się do 255, żeby dało się czytać z `serial.read()`;
 - definiowanie zmiennych (`byte`, `int`, `long`, `char`, `char*`, `char[10]`, `const char*`),
 - operatory (`+`, `-`, `*`, `/`, `mod`, `&`, `|`, `^(xor)`, `~`, `&&`, `||`, `++`, `--`),
 - przećwiczyć overflow zmiennej,
 - zapisać instrukcje (`if`, `for`, `while`, `switch`), tak aby program się skompilował
 - definiowanie funkcji `void funkcja()`; `int funkcja()`; wywołać funkcję w programie
 - funkcje (`pinMode()`, `digitalWrite()`, `delay()`), wywołać tak aby program się skompilował
 - zrobić prosty kalkulator dodający:
 - `serial.read(pierwsza_liczba)`
 - `serial.read(druga_liczba)`
 - `wynik=pierwsza_liczba+druga_liczba`
 - `serial.print(wynik)`
 - zrobić kalkulator, jw. ale dołożyć 3. `read(znak)`, potem na instrukcjach `if` zrobić odpowiednie działanie i podać wynik
 - powyższy kalkulator zrobić na `switch`
 - zrealizować mruganie LED przy pomocy instrukcji `for()`, `while()`
 - zrealizować włączanie, wyłączanie LED, uruchamiane z comy
 - do powyższego dodać opcję mrugania zrealizowaną na dwóch `delay()`: czas świecenia i czas wyłączenia z opcją ustawiania tych czasów:
 - każdy osobno
 - okres i wypełnienie
 - zrealizować mruganie LED bez `delay()`
 - zrealizować dimming LED

Lab2.

PRE:

Komputer z Windows, zainstalowane i skonfigurowane Arduino IDE, terminal, internet, pudełko z uC i czujnikami

POST:

Komunikacja z komputerem. Obsługa przez terminal. Wykresy w Kreślarce. Komunikacja pomiędzy uC.

OPIS:

- klasa Serial, funkcje begin(), print(), println(), read(), readBytesUntil(), readStringUntil(), metoda String.toInt()
- Nawiązana komunikacja obustronna z komputerem.
 - Scenariusz 1: uC pyta się o liczbę, na komputerze jest ona wpisywana, uC odpowiada kwadratem tej liczby.
- Sprawdzona w terminalu Arduino IDE i w zewnętrznym terminalu.
- Wysyłanie danych tak aby wyświetlał się wykres w Kreślarce Arduino IDE. Jedna zmienna, wiele zmiennych, legenda.
 - wysyłanie zmiennych inkrementowanych
 - wysyłanie wartości funkcji trygonometrycznej
 - wysyłanie wartości funkcji trygonometrycznej o zmieniającej się częstotliwości
- Budowa stanowiska do komunikacji między dwoma uC.
- Komunikacja dwóch uC. Praca międzygrupowa.
 - Scenariusz 1: uC-A wysyła liczbę, uC-B odbiera, mnoży przez 2 i wysyła, uC-A odbiera mnoży przez 3 i wysyła, itd. Wizualizacja na terminalu (opcja: na wykresie)
 - Scenariusz 2: uC-A odpytuje uC-B, odczytuje dane co 1s i wyświetla je na terminalu, w pewnym momencie uC-B przestaje nadawać, prosty watchdog na uC-A ma to wykryć i wysłać odpowiedni komunikat na terminal
 - Scenariusz 3: Komputer-A, podłączony do uC-A, steruje LEDem uC-B i analogicznie komputer B, podłączony do uC-B, steruje LEDem uC-A

Lab3.

PRE:

Komputer z Windows, zainstalowane i skonfigurowane Arduino IDE, terminal, internet, pudełko z uC i czujnikami

POST:

Przećwiczone: Obsługa GPIO, analog. Przerwania zewnętrzne. Input Capture.

Uruchomione: Moduł RTC, enkoder kwadraturowy, klawiatura gpio/analog, czujnik temp. ntc, czujnik akcelerometr analogowy.

OPIS:

- GPIO we/wy - szersze omówienie pinMode(), digitalWrite(), digitalRead()
- Podłączyć 2 x LED i klawiatura GPIO
- Scenariusz 1: on/off LED przy pomocy klawiatury, przycisk on i przycisk off
- Scenariusz 2: on/off LED przy pomocy klawiatury, przycisk toggle
- Scenariusz 3: on/off LED przy pomocy klawiatury, przycisk monostabilny
- Scenariusz 4: powtórzyć scenariusz 1-3 dla dwóch niezależnych LEDów - praca własna, dodać przycisk wyłączający oba LEDy jednocześnie.
- Scenariusz 5: podłączyć LED anoda do +, katoda do pinu; pokazać jak działa output z pullupem
- Scenariusz 6: podłączyć LED anoda do pinu, katoda do GND, pokazać jak działa output z pullupem a jak output push/pull, wyjaśnić dlaczego zazwyczaj zapala się LEDa sygnałem 0 a nie 1
- Scenariusz 7: input capture. pomiar czasu wciśnięcia przycisku
 - setup: attachInterrupt(digitalPinToInterrupt(pin), ISR, mode)
 - ISR(){zapis aktualnego czasu do zmiennej, wyliczenie różnicy od poprzedniego czasu i zapis do zmiennej, ustawić flagę że są nowe dane f=1}
 - main: wypisywać dane na serial jeżeli jest f=1 i zerować flagę
- Scenariusz 8: Moduł RTC, uruchomić,
 - ustawić czas,
 - odczytać bieżący czas,
 - wysyłać czas na serial co 3 sekundy
- Scenariusz 9: enkoder kwadraturowy, uruchomić na przełączniku obrotowym, zliczać impulsy i wypisywać na serial
- Scenariusz 10: analog, klawiatura analogowa, odczytać poszczególne przyciski, w zależności od wartości, wypisać odpowiedni komunikat na serial
- Scenariusz 11: analog, czujnik ntc, podłączyć, przeskalować, wypisywać na serial wartość temp (wartość surowa, przeliczona na st. C)
- Scenariusz 12: analog, akcelerometr, podłączyć, przeskalować,
 - wypisywać na serial wartość przyspieszenia 3 osi, dane surowe
 - wypisywać na serial wartość przyspieszenia 3 osi, dane przeskalowane do 1g

Lab4.

PRE:

Komputer z Windows, zainstalowane i skonfigurowane Arduino IDE, terminal, internet, pudełko z uC i czujnikami

POST:

Omówiony I2C, podłączony enkoder, IMU, wartości podane na terminal, zrobione wykresy

OPIS:

- Omówić działanie I2C, schemat podłączenia, wykres czasowy sygnału, ramka danych
- Scenariusz 1: podłączyć enkoder, odczytać wartość i wypisać na serial, zrobić wykres
- Scenariusz 2: zaproponować i zaimplementować sposób na pomiar wieloobrotowy
- Scenariusz 3: podłączyć IMU, odczytać wartości i wypisać na serial, zrobić wykres
- Scenariusz 4: zaproponować i zaimplementować sposób na pomiar wieloobrotowy
- Scenariusz 5: porównać działanie enkodera i IMU do pomiaru kąta statycznego, zbudować stanowisko, umieścić enkoder i IMU w obrotniku, zmieniać kąt i odczytywać go na enkoderze i przy pomocy IMU, zrobić wykres, zmieniać kąt powoli, szybko, zaobserwować różnice

Lab5.

PRE:

Komputer z Windows, zainstalowane i skonfigurowane Arduino IDE, terminal, internet, pudełko z uC i czujnikami

POST:

Omówiony SPI, podłączony LCD, SD, wypisywanie tekstu na LCD w dowolnym miejscu ekranu, wyświetlanie pojedynczych obrazów i sekwencyjnie (animacja), obsługa karty SD w formacie FAT32, odczytywanie, zapisywanie plików, wypisywanie listy katalogów, odczyt/zapis w podkatalogach

OPIS:

- Omówić działanie SPI, schemat podłączenia, wykres czasowy sygnału
- Scenariusz 1: Podłączyć SD, uruchomić,
 - wypisać na serial listę plików,
 - utworzyć katalog, podkatalog, utworzyć pliki w różnych miejscach,
 - zrobić listę plików/katalogów z różnych punktów (/ , /katalog/ , /katalog/podkatalog/)
 - wypisać na serial listę katalogów i podkatalogów w formie:
|katalog główny
|--plik1 w katalogu głównym
|--plik2 w katalogu głównym
|--podkatalog1
|----plik1 w podkatalog1
|----plik2 w podkatalog1
|--podkatalog2
|----plik1 w podkatalog2
|----plik2 w podkatalog2
- Scenariusz 2: obsługa plików,
 - otworzyć plik, zapisać tekst do niego,
 - odczytać plik,
 - zmienić zawartość
- Scenariusz 3: podłączyć LCD,
 - uruchomić, wyświetlić tekst, w różnych miejscach
- Scenariusz 4: podłączyć dowolny czujnik i zrobić wykres w czasie rzeczywistym na wyświetlaczu LCD

Lab6.

PRE:

Komputer z Windows, zainstalowane i skonfigurowane Arduino IDE, terminal, internet, pudełko z uC i czujnikami, puszka z powietrzem do czyszczenia komputerów

POST:

Obsłużone różne czujniki, Pulsoksymetr, EMG, czujnik CO₂, czujnik propanu (+spray)

OPIS:

- Scenariusz 1: uruchomić, odczytać, wypisać na serial, zrobić wykres dla pulsoksymetru
 - sprawdzić puls i wysycenie tlenem członków grupy (2), w różnych miejscach ciała (4),
 - Uruchomić, odczytać, wypisać na serial, zrobić wykres dla EMG, porównać wyniki
- Scenariusz 2: badania wysiłkowe
 - sprawdzić na członkach grupy (2), w różnych miejscach ciała (4, np. przedramię, ramię, lewe, prawe, udo, podudzie), porównać, wykonać pomiar w trakcie wykonywania 20 przysiadów, porównać z poprzednimi pomiarami
- Scenariusz 3: uruchomić, odczytać, wypisać na serial, zrobić wykres dla czujnika CO₂
 - sprawdzić wartość parametru w powietrzu w sali,
 - dmuchać na czujnik, zaobserwować na wykresie jaka zmienia się stężenie CO₂
- Scenariusz 4: pomiar w zamkniętej przestrzeni
 - powtórzyć pomiar zamykając czujnik w pudełku, odczytać ustabilizowaną wartość, dodać CO₂ i obserwować spadek stężenia
- Scenariusz 5: uruchomić, odczytać, wypisać na serial, zrobić wykres dla czujnika propanu
 - sprawdzić wartość parametru w powietrzu w sali, użyć sprayu, zaobserwować na wykresie jaka jest charakterystyka rozrzedzania się gazu,
- Scenariusz 6: pomiar w zamkniętej przestrzeni
 - powtórzyć pomiar zamykając czujnik w pudełku, odczytać ustabilizowaną wartość, wpuścić gaz i obserwować spadek stężenia

Lab7.

PRE:

Komputer z Windows, zainstalowane i skonfigurowane Arduino IDE, terminal, internet, pudełka z uC i czujnikami, serwo modelarskie i Dynamixel XL330-M288-T

POST:

Obsłużone serwo modelarskie i serwo Dynamixel sterowane przez protokół Protocol 2.0

OPIS:

- Scenariusz 1: podłączyć serwo modelarskie
 - ustawić pozycję zadaną z seriala
 - ustawić pozycję zadaną z seriala z małą prędkością
- Scenariusz 2: uruchomić serwo dynamixel przez komputer,
 - Podłączyć się przez Dynamixel Wizard, sterować serwem, zapoznać się z rejestrami, sterowania pozycją, prędkością, ustawić ograniczenia na pozycję, prędkości, prąd
 - sprawdzić różne sposoby sterowania serwem (pozycja, prędkość, moment), ograniczenia na położenia, prędkości, prąd - sprawdzenie jak działają
- Scenariusz 3: podłączyć serwo do uC, ustawić w serwie sterowanie pozycją
- Scenariusz 4: sterowanie bez ograniczeń, wartości wpisywane na stałe do kodu
- Scenariusz 5: sterowanie bez ograniczeń, wartości wpisywane z komputera przez serial
- Scenariusz 6: zastosować ograniczenie położenia i prędkości, sterować z serial położeniem, następnie prędkością, sprawdzić co się stanie gdy zostanie zadana wartość spoza ograniczenia
- Scenariusz 7: podłączyć akcelerometr, umieścić go na platformie. Napisać program który będzie stabilizował platformę w poziomie. Sprawdzić przechylając platformę w różnych osiach

Lab8.

PRE:

Komputer z Windows, zainstalowane i skonfigurowane Arduino IDE, terminal, internet, pudełko z uC i czujnikami

POST:

Zaprojektowany i zmontowany projekt własny

OPIS:

- Projekt własny, musi zawierać:
 - komunikację z komputerem lub zapis na kartę SD
 - odczyt min. 2 czujników
 - sterowanie serwem lub wyświetlanie na LCD
 - interakcja z użytkownikiem, przez klawiaturę lub komputer
- Zaprojektować układ:
 - schemat elektryczny
 - listing programu
 - opisać do czego może służyć
 - instrukcja uruchomienia
 - instrukcja działania

Lab9.

PRE:

Komputer z Windows, zainstalowane i skonfigurowane Arduino IDE, terminal, internet, pudełko z uC i czujnikami

POST:

zaprogramowany mikrokontroler i uruchomione poszczególne moduły

OPIS:

- Projekt własny, kontynuacja

Lab10.

PRE:

Komputer z Windows, zainstalowane i skonfigurowane Arduino IDE, terminal, internet, pudełko z uC i czujnikami

POST:

Zaprezentowany projekt własny, wystawione oceny końcowe

OPIS:

- Projekt własny, kontynuacja
- Wystawianie ocen końcowych