



Eksperymentalna ocena przydatności i możliwości płyty rozwojowej Intel Joule 570x

Experimental evaluation of the suitability and capabilities
of the Intel Joule 570x Development Kit

Michał Wasilewicz

e-mail: michal.wasilewicz.97@gmail.com

Uniwersytet Morski w Gdyni

Wydział Elektryczny

Karol Wojcieszak

e-mail: karol.wojcieszak1@gmail.com

Uniwersytet Morski w Gdyni

Wydział Elektryczny

The paper presents possibilities of Intel Joule 570x development board in teaching programming in Python and becoming familiar with basics of electronics. The main emphasis was placed on the knowledge of using operation system Linux Ostro and also practical using sensors and components from dedicate DFRobot started kit Gravity Expansion Shield for Joule V1.0. The evaluation of the usefulness and possibilities of the Intel Joule 570x Developer Kit was carried out on the generally available materials and sample programs shared on producer web page. In order to experimentally evaluate the development board under consideration, an electronic system was developed as the control object for the Intel Joule 570x board.

Key words: Intel Joule 570x, Linux Ostro, development board, programming, electronics.

Wprowadzenie

W dzisiejszych czasach tzw. płyty rozwojowe zawierające układy mikroprocesorowe odgrywają szczególną rolę w dziedzinie elektroniki oraz nauce programowania. Popularyzacja tej branży IT wynika z szerokiej dostępności różnych komponentów elektronicznych na rynku, przystępnej cenie płyt rozwojowych oraz licznych materiałów dydaktycznych udostępnianych przez producentów płyt oraz pasjonatów elektroniki i informatyki w Internecie. Szeroka oferta płyt rozwojowych (np. płyty typu Arduino) jest skierowana zarówno do amatorów, którzy chcą nauczyć się programowania oraz podstaw elektroniki, jak i do profesjonalistów, tworzących zaawansowane projekty. Producenci płyt rozwojowych udostępniają na swoich stronach dedykowane narzędzia typu IDE (*Integrated Development Environment*).

Źródłem wielu projektów pozwalających na naukę oraz zainteresowanie tematyką płyt rozwojowych jest Internet. W projektach tych przedstawiane są m.in. przykładowe zastosowania elementów elektronicznych (np. serwomechanizmów) i różnych czujników (np. akcelerometrów, czujników odległości, czujników optycznych) oraz schematy prawidłowego podłączenia tych komponentów, a także podstawy teoretyczne z nimi związane. Ponadto, również w literaturze naukowej odnaleźć można artykuły poświęcone projektom realizowanym na różnych typach płyt rozwojowych (Marciniak i inni, 201; Hatouka i inni, 2014; Zarębski i inni, 2013; Muc i inni, 2015).

W trzecim kwartale 2016 roku zaprezentowano na Intel Developer Forum platformę dedykowaną dla deweloperów zajmujących się technologią IoT (*Internet of Things*). Jest to kontynuacja wcześniejszych produktów firmy Intel. Na rynek wprowadzono dwa modele płyt rozwojowych Intel Joule, 550x i 570x. Te dwie wersje różnią się od swoich poprzedników odejściem od kompatybilności z płytami Arduino. W zamian, przyszli użytkownicy otrzymali duży skok wydajnościowy, pozwalający na tworzenie bardziej złożonych projektów m.in. wsparcie dla strumieni danych wideo w formacie 4K, możliwość obróbki danych związanych z procesem skanowania trójwymiarowego 3D, możliwość obróbki danych związanych z rozpoznawaniem obiektów i obrazów dzięki zastosowaniu kamer RealSense. Zaproponowane przez Intel płyty umożliwiają także realizację takich projektów jak tworzenie robotów, czy też konstruowanie dronów (Lehrbaum, 2016).

Pomimo wielu możliwości wykorzystania płyty rozwojowej Intel Joule 570x, została ona wycofana z rynku po niespełna roku. Spowodowane to było przede wszystkim wysokimi kosztami zakupu płyty (w chwili wypuszczenia na rynek płyta kosztowała

wała ok. 370 \$), co jest wysoką ceną dla wielu projektantów (szczególnie studentów) oraz wysokim stopniem skomplikowania konfiguracji tej płytki rozwojowej.

W niniejszej pracy podjęto próbę przedstawienia na drodze eksperymentalnej możliwości płyty rozwojowej Intel Joule 570x i jej przydatności w nauce programowania oraz tworzenia projektów w obszarze elektroniki i techniki mikroprocesorowej.

Opis płyty rozwojowej Intel Joule 570x i dostępnych dla płyty akcesoriów

Płyta rozwojowa Intel Joule 570x opiera się na 64-bitowym procesorze typu SoC (*System on Chip*) Intel Atom Broxton-M T5700, wyposażonym w cztery rdzenie o częstotliwości 1,7 GHz (lub 2,4 GHz z opcją turbo). Mikroprocesor wspierany jest przez pamięć RAM LPDDR4 o pojemności 4 GB oraz pamięć eMMC o pojemności 16 GB. Płyta posiada kartę graficzną Intel Gen9LP HD Graphics ze wsparciem dla strumieni wideo w jakości 4K. Do komunikacji bezprzewodowej Intel Joule 570x wykorzystuje dwa moduły: WiFi 802.11 ac oraz Bluetooth 4.1. Dodatkowo, przyszli użytkownicy mają możliwość pracy na preinstalowanym systemie operacyjnym Linux Ostro. Oprócz systemu Linux Ostro, dedykowanymi systemami operacyjnymi dla tej płyty są Ubuntu 16.04 LTS oraz Windows 10 IoT. Firma Intel charakteryzuje rozważaną płytę jako zaawansowaną platformę obliczeniową, przystosowaną do nowej generacji inteligentnych urządzeń. Na rysunku 1 (Lehrbaum, 2016) przedstawiono wygląd modułu procesora Intel Atom Broxton-M T5700. Jak widać, na górnej stronie płyty zainstalowano moduły WiFi oraz Bluetooth, a także układ scalony zarządzający zasilaniem PMIC (*Power Management Integrated Circuit*). Natomiast na dolnej stronie znajdują się 2 porty MIPI CSI służące do podłączenia kamery, pamięć masowa eMMC, na której znajduje się domyślnie preinstalowany system operacyjny Linux Ostro oraz dwa interfejsy służące do komunikacji z elementami płyty montażowej Tuchuck Intel Joule 570x (Nota katalogowa, Intel Joule Datasheet August 2016 Revision 1.0).

Moduł procesorowy przymocowany jest do płytki montażowej Tuchuck, składającej się z takich elementów jak: gniazdo zasilające DC Jack 2,5 – 5,0 V, port USB-C, port USB 3.0, port USB Serial, port Micro – HDMI, interfejs służący do podłączenia modułu procesorowego, slot na kartę Micro – SD, złącze wentylatora, gniazdo na baterię zegara czasu rzeczywistego, trzy przyciski, dwa interfejsy MPI CSI, dwa interfejsy 40 – pinowe służące do podłączenia elementów elektronicznych (nota katalogowa, Intel Joule Datasheet August 2016 Revision 1.0). Na rysunku 2 (Lehrbaum,

2016) pokazano wygląd płytki montażowej.

Intel Joule 570x, jak większość płytek rozwojowych dostępnych na rynku, posiada dedykowany temu modelowi zestaw czujników. W celu wykorzystania potencjału tej płyty rozwojowej, producent DFRobot stworzył zestaw sensorów *Gravity Sensor Kit* - rysunek 3 (strona internetowa producenta DFRobot), wraz z nakładką *Gravity Expansion Shield for Joule V1.0* pasującej do oryginalnej płyty rozwojowej Intel Joule 570x. Nakładka ta pozwala na podłączenie kilku czujników naraz. Na rysunku 4 przedstawiono rozmieszczenie pinów na nakładce wraz z ich opisami oraz numerami. Do dyspozycji projektantów są porty analogowe oraz porty cyfrowe, w tym pozwalające na przedstawienie zastosowania modulacji PWM oraz interfejsów typu SPI, UART i I²C (dokumentacja techniczna płyty rozszerzeń Gravity Expansion Shield for Intel Joule).

Projekt układu testowego

W projekcie umożliwiającym eksperymentalną ocenę płyty Intel Joule 570x wykorzystano język programowania Python, preinstalowany system operacyjny Linux Ostro oraz czujniki z zestawu Gravity Sensor Kit. Należy w tym miejscu zwrócić uwagę na fakt, iż przeszkodą dla projektantów pragnących zastosować w praktyce rozważaną w pracy płytę ewaluacyjną, jest mała dostępność gotowych już projektów, na podstawie których można oprzeć realizację własnych układów i systemów.

Na rysunku 5 przedstawiono schemat blokowy opracowanego układu, natomiast na rysunku 6 pokazano zdjęcie rzeczywistego układu. Jak widać na rysunkach, płyta rozwojowa połączona jest z dziewięcioma układami peryferyjnymi: 1 – ADKeyboard, 2 – czujnik wilgotności, 3 – czujnik natężenia światła, 4 – czujnik temperatury, 5 – regulowany potencjometr, 6 – przetwornik akustyczny piezoelektryczny, 7 – niebieska dioda LED, 8 – zielona dioda LED, 9 – czerwona dioda LED. Układy od 1 do 5 połączone są z portem analogowym, z kolei układy od 6 do 9 połączono z portem cyfrowym na płycie Gravity Expansion Shield for Intel Joule.

Algorytm pracy układów testowych i opracowane kody programów

Działanie układu testowego jest zgodne z algorytmem z rysunku 7. Wybór określonego przycisku z układu 1 skutkuje wybraniem jednej z pięciu operacji. Każda z tych operacji (procesów pomiarowo-sterujących) może być przerwana poprzez ak-

tywając przycisku przydzielonego do innej operacji.

Należy zaznaczyć, że aby napisać kod programu, należy skorzystać z wbudowanego w system operacyjny Linux Ostro edytora tekstu VI. W tym celu należy użyć komendy *vi nazwa_pliku.py*. Pozwala ona na utworzenie pliku wykonawczego, który można edytować. Po napisaniu kodu należy zapisać plik wychodząc z trybu edycji używając sekwencji <Esc> i wpisując komendę :wq!. Do skompilowania i uruchomienia napisanego programu należy użyć komendy *python nazwa_pliku.py*. Natomiast aby zatrzymać program należy wcisnąć sekwencję klawiszy na klawiaturze <Ctrl> + <Z>.

Na stronie internetowej firmy Intel poświęconej płytce rozwojowej Intel Joule 570x można znaleźć przykładowy kod napisany w języku Python. Na listingu z okna nr 1 przedstawiono podstawowy przykładowy kod programu pozwalający na sterowanie diodami LED zainstalowanymi na płycie montażowej Tuchuck. Przykładowy program pokazany w oknie 1 używa bibliotek *mraa* oraz *time*. Pierwsza z nich służy do komunikacji z podłączonymi komponentami elektronicznymi, druga natomiast pozwala na wprowadzenie opóźnień między wykonywanymi operacjami. W funkcji programu *main()* przypisano do zmiennej *leds* cztery diody znajdujące się na płycie montażowej, każda z nich posiada unikalny numer portu. Następnie w pierwszej pętli *for* przypisuje się zbiór diod LED ze zmiennej *leds* na wyjściu za pomocą funkcji *led.dir(mraa.DIR_OUT)*. W pętli *while* program załącza i wyłącza kolejno każdą diodę oraz pojawia się komunikat dla danej diody o jej stanie. Program wykonuje się w nieskończoność do momentu jego przerwania. Rozważany program jest prostym kodem, pozwalającym na wstępne zapoznanie się z językiem programowania Python dla omawianej płyty rozwojowej.

Opracowany przez autorów kod programu testowego pokazany jest na listingach w oknach od numeru 2 do 9. Należy zaznaczyć, że odczyt informacji (sygnałów) z podłączonych komponentów elektronicznych realizowany jest przez przetwornik analogowo-cyfrowy ADS1115. Jak widać w listingu 2, analogicznie do przykładowego programu z listingu 1, napisany program korzysta z bibliotek *mraa* oraz *time*. W funkcji *main()* wykonywany jest program, który określa jaka operacja będzie wykonana po wciśnięciu jednego z pięciu przycisków na klawiaturze. Na początku w kodzie programu w funkcji *main()* zdefiniowano porty dla diod LED, brzęczka, klawiatury, czujnika wilgotności, czujnika temperatury, czujnika natężenia światła oraz pokręta. W pętli *while* program odczytuje stan klawiatury. Stwierdzenie wciśnięcia przycisku skutkuje przejściem z użyciem warunku *if* do określonego fragmentu kodu

(poszczególne listingi), obsługującego wybrany układ peryferyjny. Wybór przycisku bazuje na określonej wartości napięcia, mierzonej przez przetwornik analogowo-cyfrowy ADS1115 umieszczony na płycie rozszerzeń, przypisanej do danego przycisku. Dla potrzeb programu wartości przycisków wyznaczono w testach metodą analizy wartości uzyskiwanych po wciśnięciu danego przycisku. Jak można zauważyć, wartości przypisane do przycisków mieszczą się w określonych zakresach. Wynika to z faktu, że wartość uzyskiwana po wciśnięciu określonego przycisku w wyniku konwersji analogowo-cyfrowej nie jest stała i charakteryzuje się pewną zmiennością. W opracowanym programie odczyt stanu klawiatury jest realizowany w sposób ciągły, przy czym fragment kodu przypisany do określonego przycisku jest wykonywany do momentu wciśnięcia innego przycisku.

Na listingu w oknie nr 3 przedstawiono działanie programu po wciśnięciu przycisku S1. Program odczytuje oraz zapisuje w rejestrze wartości odczytane z czujnika wilgotności. Następnie program wyznacza różnicę między wilgotnością dla 100% a bieżącą wartością odczytaną. Wartość zmiennej *humidity0* równą 13600 wyznaczono na bazie wcześniejszych testów tego czujnika (zgodnie z metodą producenta czujnika). W programie wyznaczona wilgotność jest przeliczana na wartość podawana w [%]. W rozważanym fragmencie kodu programu wykorzystywane są trzy diody LED: czerwona, zielona oraz niebieska. W pętlach *if-elif* zdefiniowano stan, w którym dana dioda załączy się. Przykładowo, jeśli wilgotność gleby będzie większa lub równa 70%, to załączy się dioda niebieska.

Na listingu w oknie nr 4 przedstawiono działanie programu po wciśnięciu przycisku S2. Program w tym fragmencie kodu odczytuje temperaturę otoczenia. W dokumentacji technicznej udostępnionej przez producenta czujnika temperatury można znaleźć informację, że czujnik ten mierzy wartości w zakresie od 0 do 150°C z dokładnością $\pm 1^\circ\text{C}$. Program odczytuje i zapisuje cyfrową wartość odpowiadającą danej temperaturze otoczenia, a następnie następuje konwersja tej wartości na stopnie Celsjusza.

Na listingu z okna nr 5 zamieszczono kod programu realizowany po wciśnięciu przycisku S3. W tym przypadku płyta rozwojowa odczytuje natężenie oświetlenia z wykorzystaniem detektora optycznego, jakim jest czujnik DFR0026. W dokumentacji technicznej udostępnionej przez producenta można znaleźć informację, że czujnik ten mierzy natężenie światła w zakresie od 1 do 6000 lx. Podobnie jak w przypadku detektora temperatury odczytywana wartość cyfrowa jest konwertowana na jednostki natężenia światła.

Z kolei na listingu z okna nr 6 przedstawiono działanie programu po wciśnięciu przycisku S4. Program w tym przypadku odczytuje pozycję pokrętki potencjometru. W dokumentacji technicznej udostępnionej przez producenta elementu elektronicznego można znaleźć informację, że regulacja kątowa obrotu pokrętki mieści się w zakresie od 0 do 300 °. Odczytywana wartość cyfrowa konwertowana jest następnie na stopnie.

Na listingu z okna nr 7 przedstawiono działanie programu po wciśnięciu przycisku S5. Program uruchamia tzw. brzęczek (przetwornik piezoelektryczny). Dźwięk jest generowany z częstotliwością 1 Hz.

Listing z okna nr 8 prezentuje ostatnią linię kodu. Linia ta służy do wykonywania kodu tylko wtedy, gdy plik został uruchomiony bezpośrednio, a nie został zaimportowany.

Na listingu z okna nr 9 można zauważyć kod służący do przerywania pętli *if-elif*. Przy wykryciu napięcia odpowiadającego wciśnięciu odpowiedniego przycisku na klawiaturze DFR0075, następuje przerwanie wykonywanego przez program zadania i przejście do operacji wybranej przyciskiem, co pokazano na rysunku nr 8, przedstawiającym moment, w którym przerwano wyświetlanie temperatury na rzecz wyświetlenia wartości natężenia światła.

Analiza i ocena działania układu opartego na Intel Joule 570x

Opracowany program testowy pozwolił przetestować możliwości rozważanej płyty rozwojowej pod kątem jej przydatności w zastosowaniach kontrolno-pomiarowych. Należy podkreślić, że układ elektroniczny oparty na Intel Joule 570x działał zgodnie z napisanym kodem programu. Po wciśnięciu określonych przycisków układ wykonywał zadane operacje zapisane w kodzie, zgodnie i z uwzględnieniem specyfiki działania poszczególnych komponentów zastosowanych w układzie. Można zatem stwierdzić, że rozważana w pracy płyta rozwojowa stanowi dobrą bazę, jako układ zbierający i przetwarzający informację cyfrową z różnych typów sensorów. Niewątpliwie, płyta ta może być z powodzeniem stosowana jako główne urządzenie sterujące w zaawansowanych projektach z dziedziny robotyki czy informatyki.

Największą wadą w wykonywanym projekcie, weryfikującym użyteczność płyty, był brak środowiska programistycznego dedykowanego płycie rozwojowej Intel Joule 570x pozwalającego na pisanie programów w języku Python. Aby usprawnić pisanie programu, należało napisać kod w innym dedykowanym programie, np. Py-

Charm lub IDLE przed wklejeniem go do pliku w edytorze VI. Wynika to z tego, że wyżej wymienione programy poprawiały automatycznie składnię napisanego kodu, co nie jest możliwe w edytorze VI. Niedogodność ta nie jest uciążliwa, gdy pisze się programy zawierające małą ilość linii kodu. Natomiast dla większych i bardziej rozbudowanych projektów, po przekopiowaniu wcześniej przygotowanego programu, należy dokładnie przeanalizować i skorygować skopiowaną zawartość, ze względu na złe formatowanie tekstu.

Zdaniem autorów, płyta rozwojowa Intel Joule 570x nie jest przeznaczona dla początkujących projektantów układów elektronicznych. Praktyczne wykorzystanie tej płyty związane jest z koniecznością posiadania wiedzy teoretycznej oraz praktycznej z zakresu elektroniki, języków programowania oraz programowania układów mikroprocesorowych, np. mikrokontrolerów, co najmniej na poziomie średnim. Obecnie firma Intel wycofała swoje wsparcie dla tej płyty rozwojowej. Z tego powodu zakupienie dedykowanego zestawu czujników jak i samej płyty rozwojowej może stanowić problem. Wynika to m.in. z ich małej dostępności na polskim rynku urządzeń elektronicznych. Znacznie bardziej przyjaznymi płytami ewaluacyjnymi są np. płyty z serii Arduino.

Podsumowanie

Niniejszy artykuł przedstawia ocenę możliwości wykorzystania i zastosowania płyty rozwojowej Intel Joule 570x do tworzenia programów sterujących układami czujników pomiarowych (sensorów). Jak stwierdzono eksperymentalnie, rozważana płyta rozwojowa oferuje znaczne możliwości w zakresie zbierania i obróbki danych cyfrowych, np. w projektach typu Embedded oraz IoT. W dalszych działaniach autorów związanych z niniejszą pracą planowane jest rozwinięcie sieci sensorów, dodanie funkcji sterowania poszczególnymi sensorami oraz stworzenie aplikacji internetowej do zarządzania taką siecią.

Podziękowania

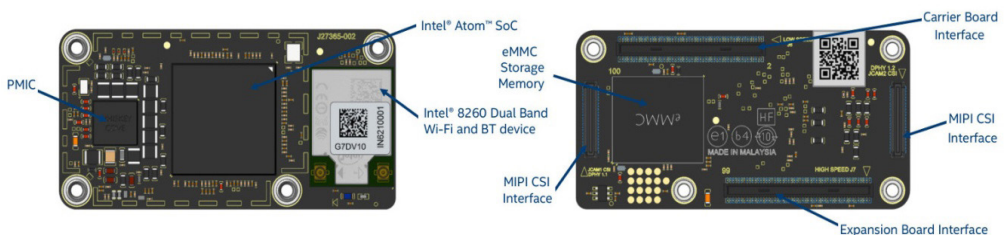
Autorzy pracy składają podziękowania dr inż. Jackowi Dąbrowskiemu z Katedry Elektroniki Morskiej z Uniwersytetu Morskiego w Gdyni za opiekę naukową i wsparcie merytoryczne podczas redakcji niniejszego artykułu.

Bibliografia

- Dokumentacja techniczna płyty rozszerzeń Gravity Expansion Shield for Intel Joule.*
- Hatouka, R. Kamiński, M. (2015). Zastosowanie modułu Arduino w układzie zdalnego sterowania robotem mobilnym. *Prace Naukowe Instytutu Maszyn, Napędów i Pomiarów Elektrycznych Politechniki Wrocławskiej, nr 70, ss. 121-132.*
<http://linuxgizmos.com/intel-debuts-joule-iot-platform-with-ostro-linux-support/>
https://wiki.dfrobot.com/Gravity_Expansion_Shield_for_Intel_Joule_SKU__DFR0465#target_9;
https://www.dfrobot.com/index.php?route=product/product&product_id=1561
[https://www.mouser.com/datasheet/2/612/intel-joule-module-datasheet-1018643-1216094.pdf/](https://www.mouser.com/datasheet/2/612/intel-joule-module-datasheet-1018643-1216094.pdf)
- Lehrbaum, R. (2016). *Tiny Intel „Joule” IoT module runs Ostro Linux on quad-core Atom.*
- Marciniak, T. Dąbrowski, A. Puchalski, R. Dratwiak, D. Marciniak, W. (2019). Zastosowanie mikrokontrolera STM32F410 do prezentacji zagadnień cyfrowego przetwarzania sygnałów. *Przegląd Elektrotechniczny, R. 95, nr 10/2019, ss. 118-120.*
- Muc, A. Murawski, L. Gesella, G. Szeleziński, A. Szarmach, A. (2015). Zastosowanie platform cyfrowych Arduino i Raspberry Pi w nauczaniu sterowania obiektem pneumatycznym. *Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej, 41/205, ss. 45-50.*
- Nota katalogowa, Intel Joule Datasheet August 2016 Revision 1.0.*
- Strona internetowa producenta DFRobot.*
- Zarębski, J., Górecki, K. Dąbrowski, J. (2013). *Measuring System for Determining Thermal Parameters of Semiconductor Devices.* Międzynarodowa Konferencja MIXDES 2013, 316-320

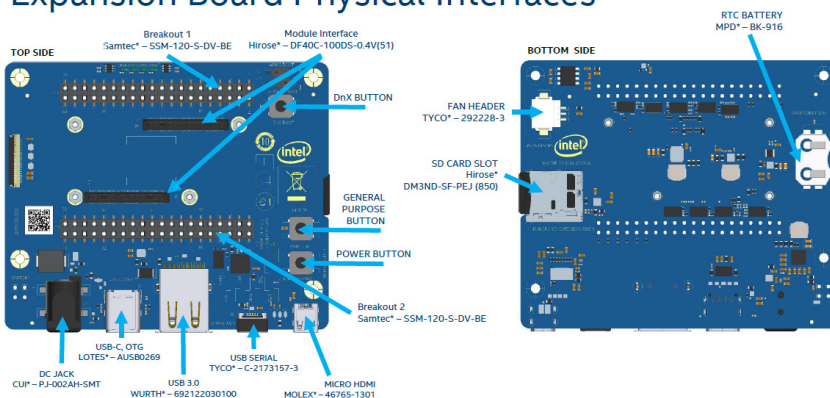
Rysunek 1

Budowa modułu Intel Joule 570x



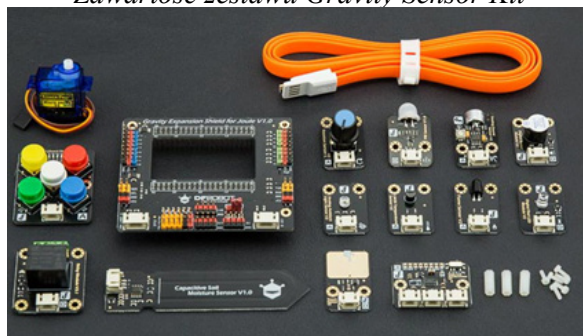
Rysunek 2

Schemat interfejsów płyty montażowej Tuchuck Intel Joule 570x
Expansion Board Physical Interfaces



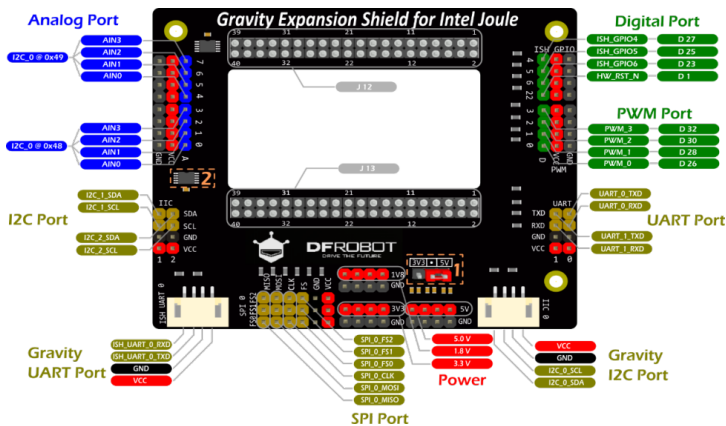
Rysunek 3

Zawartość zestawu Gravity Sensor Kit



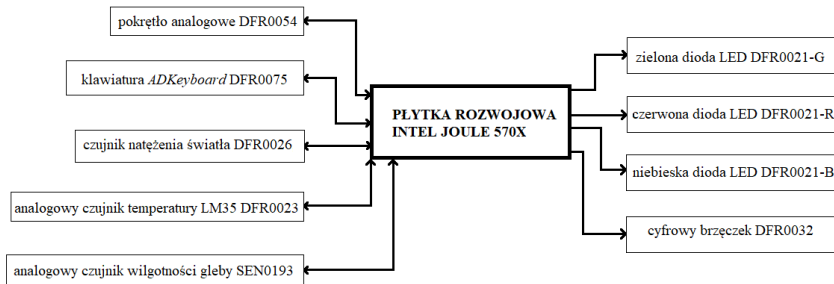
Rysunek 4

Schemat nakładki do oryginalnej płyty rozszerzeń Gravity Expansion Shield dla Intel Joule



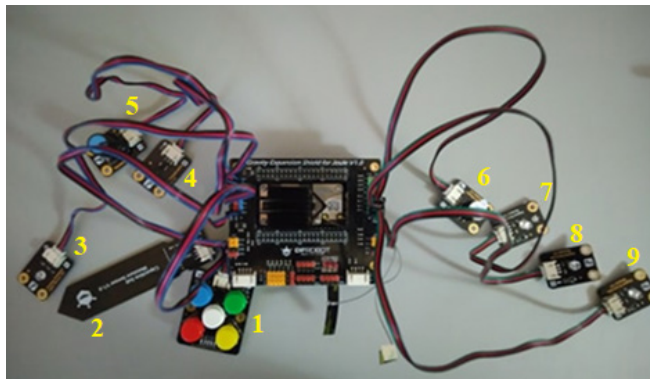
Rysunek 5

Schemat blokowy układu testowego



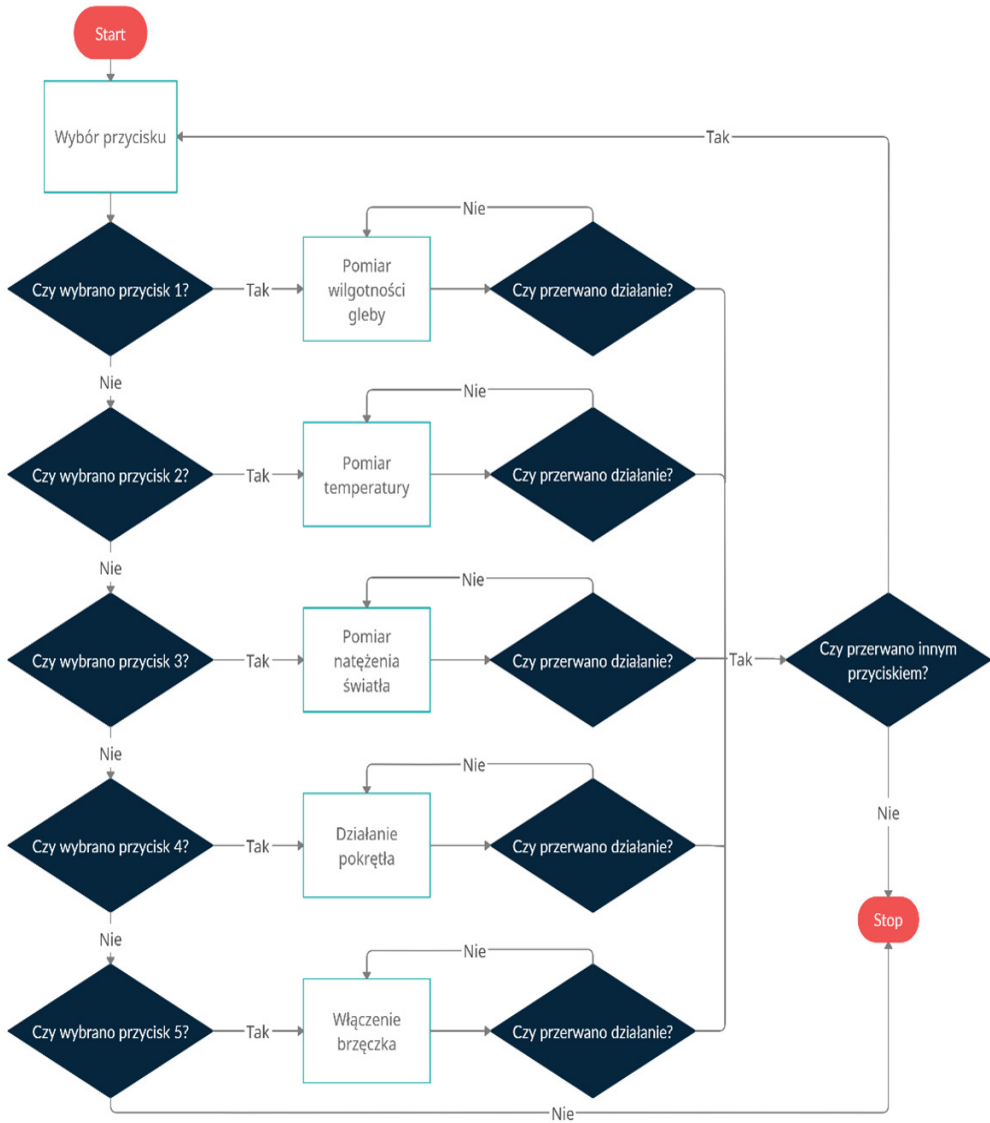
Rysunek 6

Układ testowy



Układ testowy: 1 – ADKeyboard, 2 – czujnik wilgotności, 3 – czujnik natężenia światła, 4 – czujnik temperatury, 5 – regulowany potencjometr, 6 – przetwornik akustyczny piezoelektryczny, 7 – niebieska dioda LED, 8 – zielona dioda LED, 9 – czerwona dioda LED

Rysunek 7
Algorytm działania programu



Rysunek 8

Przykład przerwania bieżącej operacji innym przyciskiem (przerwanie operacji wywołanej przyciskiem S2 za pomocą przycisku S3)

```

root@intel-corei7-64:~# python projekt.py
Temperature: 21.91
Temperature: 21.78
Temperature: 21.76
Temperature: 21.73
Temperature: 21.71
Temperature: 21.69
Temperature: 21.67
Temperature: 21.66
Temperature: 21.62
Lux: 263.5
Lux: 71.5
Lux: 70.4
Lux: 69.2
Lux: 71.5
Lux: 73.2
Lux: 69.0
Lux: 70.6
^Z
[3]+  Stopped(SIGTSTP)                python projekt.py

```

Okno 1. Przykładowy kod programu (listing) z oficjalnej strony internetowej firmy Intel

```

import mraa
import time

def main():
    leds = [mraa.Gpio(100), mraa.Gpio(101), mraa.Gpio(102), mraa.Gpio(103)]
    for led in leds:
        led.dir(mraa.DIR_OUT)
    while(1):
        for led in leds:
            led.write(1)
            print("LED" + str(leds.index(led))+" status: " + str(led.read()))
            time.sleep(0.3)
            led.write(0)
            print("LED" + str(leds.index(led))+" status: " + str(led.read()))

if __name__ == "__main__": main()

```

Okno 2. Konfiguracja bibliotek, przypisanie elementów do portów oraz początek pętli while True

```
import mraa
import time

def main():
    #PWM
    redLed = mraa.Gpio(26)
    greenLed = mraa.Gpio(28)
    blueLed = mraa.Gpio(30)
    buzzer = mraa.Gpio(32)

    #Ustawienie elementow podlaczonych pod PWM na wyjście
    redLed.dir(mraa.DIR_OUT)
    greenLed.dir(mraa.DIR_OUT)
    blueLed.dir(mraa.DIR_OUT)
    buzzer.dir(mraa.DIR_OUT)

    #ADKeyboard
    keyboard = mraa.I2c(0)
    keyboard.address(0x48)

    #Czujnik wilgotności
    humidity0 = 13600
    hum = mraa.I2c(0)
    hum.address(0x49)

    #Czujnik temperatury
    temp = mraa.I2c(0)
    temp.address(0x49)

    #Czujnik natężenia światła
    light = mraa.I2c(0)
    light.address(0x49)

    #Pokretło
    rotation = mraa.I2c(0)
    rotation.address(0x49)

    while True:
        #odczyt/zapis wartości w chwili braku wciśnięcia przycisków
        keyboard.writeWordReg(1, 0x83C1)
        val = keyboard.readWordReg(0)
        analogValueX = ((val & 0xff00) >> 8) + ((val & 0x00ff) << 8)
```

Okno 3. Kod programu dla przycisku S1

```
# S1 - czujnik wilgotności
if ((analogValueX >= 14200) & (analogValueX <= 14250)):
    while (1):
        # odczyt/zapis wilgotności w systemie dziesiętnym
        hum.writeWordReg(1, 0x83E1)
        soilMoistureValue = hum.readWordReg(0)
        analogValueH = ((soilMoistureValue & 0xff00) >> 8) + ((soilMoistureValue & 0x00ff) << 8)
        # Przeliczanie wartości wilgotności na wartość w %
        roznica = humidity0 - analogValueH
        hpercent = abs((roznica / 7100.0) * 100)
        if ((hpercent >= 0) & (hpercent <= 100)):
            if (hpercent >= 70):
                redLed.write(0)
                greenLed.write(0)
                blueLed.write(1)
                print("High Humidity: " + str(round(hpercent, 1)) + "%")
                time.sleep(1)
            elif ((hpercent < 70) & (hpercent > 30)):
                redLed.write(0)
                greenLed.write(1)
                blueLed.write(0)
                print("Medium Humidity: " + str(round(hpercent, 1)) + "%")
                time.sleep(1)
            elif (hpercent <= 30):
                redLed.write(1)
                greenLed.write(0)
                blueLed.write(0)
                print("Low Humidity: " + str(round(hpercent, 1)) + "%")
                time.sleep(1)
```

Okno 4. Kod programu dla przycisku S2

```
# S2 - czujnik temperatury
elif ((analogValueX >= 16200) & (analogValueX <= 16250)):
    while (1):
        # odczyt/zapis temperatury w systemie dziesiętnym
        temp.writeWordReg(1, 0x83C1)
        t = temp.readWordReg(0)
        analogValueT = ((t & 0xff00) >> 8) + ((t & 0x00ff) << 8)
        # konwersja z wartości dziesiętnej na stopnie Celsjusza
        temperature = (analogValueT * 3300) / 192000.0
        print("Temperature: " + str(round(temperature, 2)) + " C")
        # odczyt/zapis wartości w chwili braku wciśnięcia przycisków
        keyboard.writeWordReg(1, 0x83C1)
        val = keyboard.readWordReg(0)
        analogValueX = ((val & 0xff00) >> 8) + ((val & 0x00ff) << 8)
        time.sleep(1)
```

Okno 5. Kod programu dla przycisku S3

```
# S3 - czujnik natężenia światła
elif ((analogValueX >= 18190) & (analogValueX <= 18240)):
    while (1):
        # odczyt/zapis natężenia światła w systemie dziesiętnym
        light.writeWordReg(1, 0x83D1)
        l = light.readWordReg(0)
        analogValueL = ((l & 0xff00) >> 8) + ((l & 0x00ff) << 8)
        # konwersja z wartości dziesiętnej na luxy
        lux = analogValueL / 4.77
        print("Lux: " + str(round(lux, 1)))
```

Okno 6. Kod programu dla przycisku S4

```
# S4- pokretło
elif ((analogValueX >= 20890) & (analogValueX <= 20940)):
    while (1):
        # odczyt/zapis pozycji pokretła w systemie dziesiętnym
        rotation.writeWordReg(1, 0x83F1)
        rot = rotation.readWordReg(0)
        analogValueR = ((rot & 0xff00) >> 8) + ((rot & 0x00ff) << 8)
        degree = (analogValueR * 300) / 28672

        # konwersja z wartości dziesiętnej na stopnie
        if ((degree >= 0) & (degree <= 300)):
            print("Degree: " + str(degree))
```

Okno 7. Kod programu dla przycisku S5

```
# S5 - buzzer
elif ((analogValueX >= 24410) & (analogValueX <= 24460)):
    while (1):
        # buzzer on
        buzzer.write(1)
        time.sleep(1)
        # buzzer off
        buzzer.write(0)
```

Okno 8. Warunek wykonania funkcji main().

```
if __name__ == "__main__": main()
```

Okno 9. Przykładowe przerwanie przyciskiem danej operacji

```
if (((analogValueX >= 14200) & (analogValueX <= 14250))):
    break
elif (((analogValueX >= 16200) & (analogValueX <= 16250))):
    break
elif (((analogValueX >= 18190) & (analogValueX <= 18240))):
    break
elif (((analogValueX >= 20890) & (analogValueX <= 20940))):
    break
```