



Projekt z przedmiotu
Komputerowe Techniki
Pomiarowe
Bartłomiej Sarata

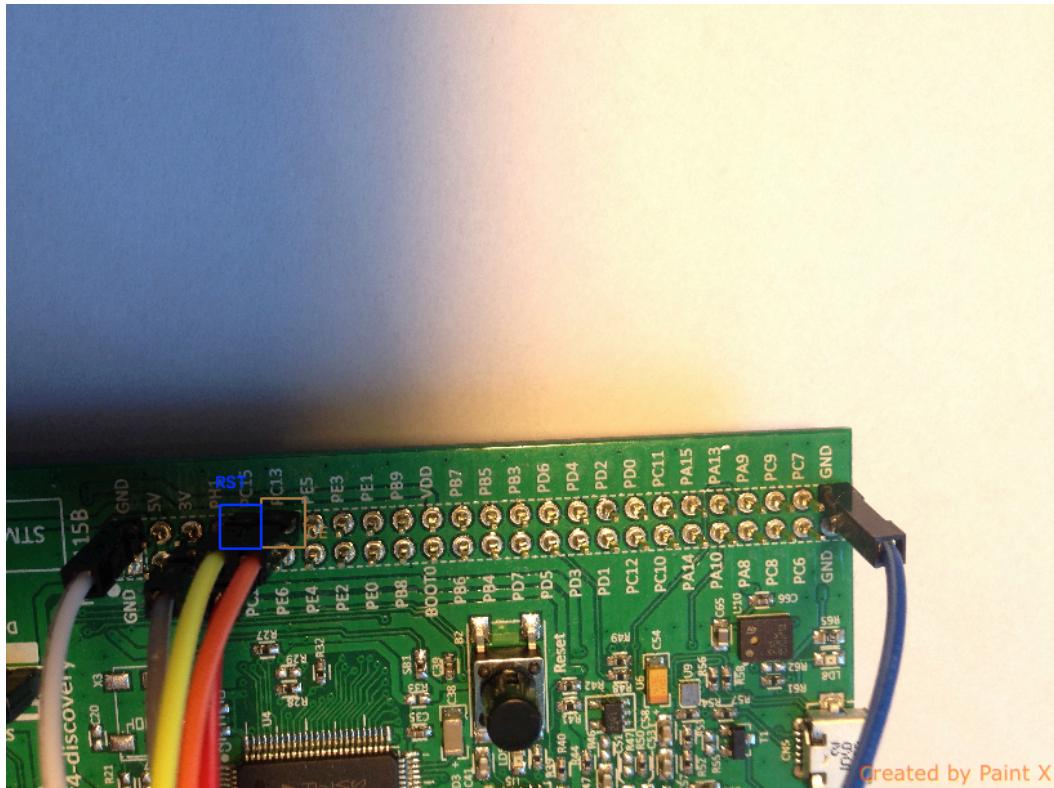
Na płytce STM32F4-401C-DISCO z wyświetlaczem LCD NOKIA5110 3V-5V oraz potencjometrem 10 om.

Schemat podpięcia pinów Płytki STM32F4 oraz wyświetlacza LCD NOKIA5110

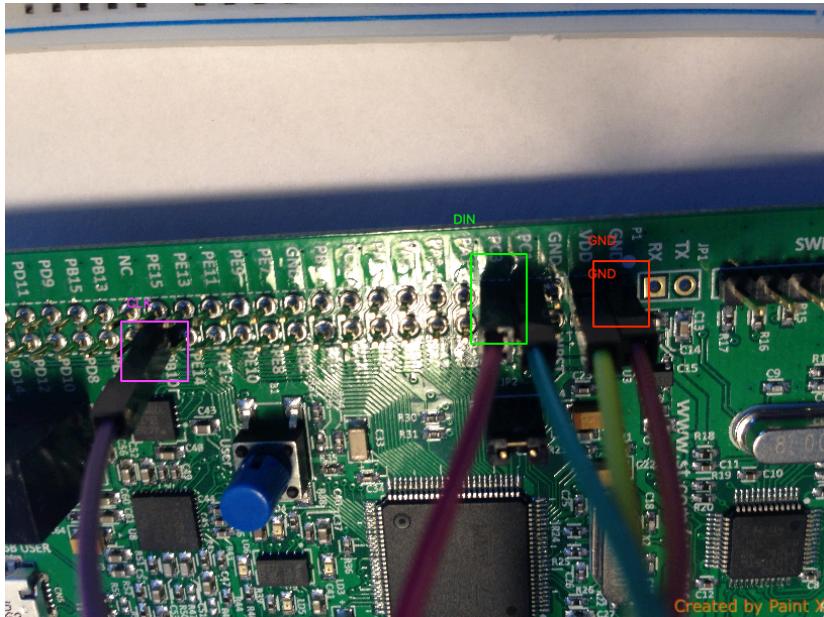
WYŚWIETLACZ LCD	PŁYTKA STM32	OPIS
RST	PC15	PIN RESETUJĄCY LCD
CE	PC13	WŁĄCZENIE CHIPU SPI2
DC	PC14	PIN DANE/KOMENDY
DIN	PC3	Master Output, Slave Input PIN DLA SPI2
CLK	PB10	PIN ZEGARU DLA SPI2
VCC	3.3V	ZASILANIE LCD
LIGHT	GND	JEŻELI ZOSTANIE PODŁĄCZONY POD GND, CZARNE ŚWIATŁO JEST WŁĄCZONE
GND	GND	UZIEMIENIE

Zdjęcie pokazujące schemat w rzeczywistości:

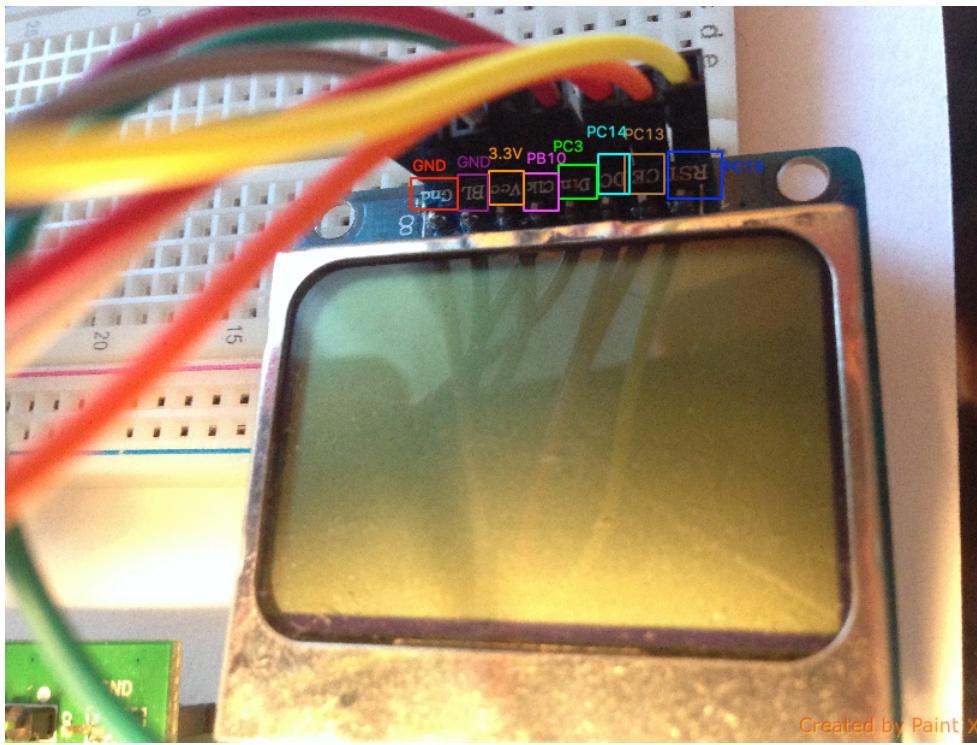
Płytki STM32F4 cz.1:



Płytki STM32F4 cz.2:



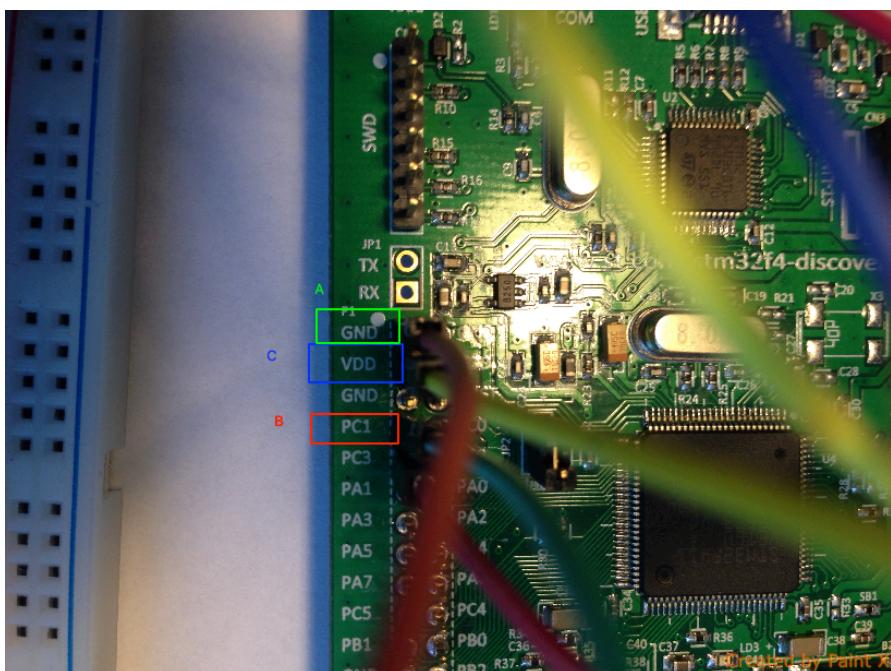
Płytki stykowa z LCD:



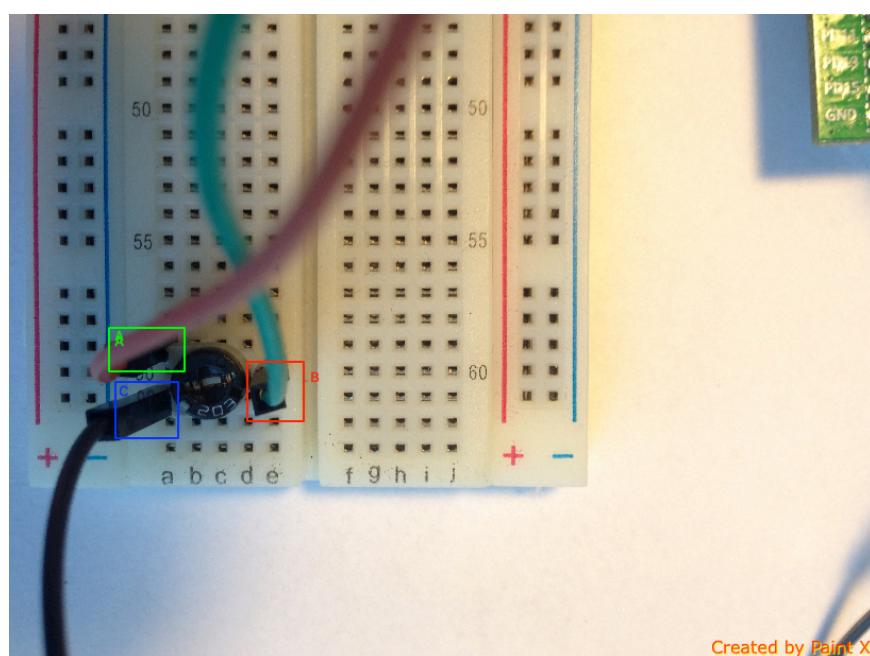
Schemat podpięcia pinów Płytki SMT32F4 oraz potencjometru 10 OHM/20 OHM

Potencjometr	PŁYTKA STM32	OPIS
A 0V	GND	UZIEMIENIE POTENCJOMETRU
C 3.3V	VDD	ZASILANIE POTENCJOMETRU
B ADC INPUT	PC2	WYSYŁANIE SYGNAŁU Z POTENCJOMETRU

Zdjęcie pokazujące schemat w rzeczywistości:



Płytki stykowa z potencjometrem:



W projekcie wykorzystałem istniejące biblioteki obsługujące wyświetlacz LCD NOKIA5110. Ten wyświetlacz był używany w telefonie nokia 5110/3310. Jest tani i bardzo powszechny w użyciu:
Właściwości wyświetlacza:

- Rozdzielcość: 84x84px
- Biblioteka graficzna dla linii, prostokątów oraz kół,
- Algorytm do odświeżania tylko zmienianego pola
- Sterownik z SPI
- Programowa zmiana kontrastu

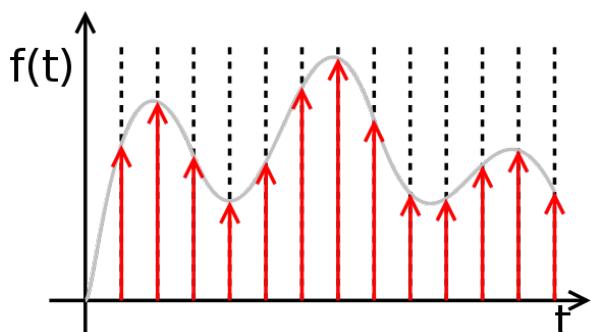
Przetwornik analogowo-cyfrowy ADC

ADC (ang. Analog to Digital Converter) jest jednym z podstawowych peryferiów będącym na wyposażeniu mikrokontrolera STM32F4. Dokonuje on konwersji wartości napięcia sygnału analogowego na postać cyfrową.

Dyskretyzacja

Sygnał analogowy jest sygnałem ciągłym. Oznacza to tyle, że między dwiema dowolnie bliskimi chwilami zawsze istnieje trzecia, w której sygnał również istnieje i przyjmuje pewną wartość. Nie jesteśmy zmierzyć wszystkich wartości.

Cyfrowe urządzenie pomiarowe- przetwornik ADC, ma pewną częstotliwość graniczną, z którą jest w stanie wykonywać pomiary. Oznacza to tyle, że istnieje określony czas pomiędzy następującymi po sobie pomiarami, **w którym nie jesteśmy w stanie wykonać kolejnego pomiaru**.



Czerwone strzałki to chwile dokonania pomiarów. Pomiędzy nimi sygnał analogowy (oznaczony szarą linią) istnieje i przyjmuje konkretne wartości. Wszystkie informacje o poziomie sygnału znajdujące się pomiędzy pomiarami zostają utracone. Taki proces nazywamy dyskretyzacją (zamiana czasu sygnału z ciągłego na dyskretny).

Kwantyzacja

Sygnał analogowy charakteryzuje się wartością o nieskończonej dokładności. Pomiędzy dwiema dowolnie bliskimi wartościami poziomu napięcia istnieje trzecia, którą sygnał może przyjąć.

The screenshot shows the Eclipse IDE interface with the main.c file open. The code is written in C and performs the following tasks:

- Initialization of variables: zrodlo[20], handler = 0, i = 0.
- Initialization of functions: ADCInit(), DMAInit(), TM_DELAY_Init(), GPIOInit_Nokia(), SPIInit_Nokia(), NokiaInit().
- Cleaning the LCD screen: LCDClear().
- Looping while (1):
 - Drawing "PROJEKT KP" at position (1, 1).
 - Reading 10 values from ADC into handler.
 - Average calculation: handler = (handler / 10);
 - Displaying the result: sprintf(zrodlo, "%d", handler); LCDDraw(3, 1, zrodlo);
 - Displaying voltage: sprintf(zrodlo, "V:%.2f", ((3.3 / 4096.0) * (handler))); LCDDraw(4, 1, zrodlo);
- Resetting handler to 0.
- Delay of 500ms.
- Final LCD clear.

Kod odpowiedzialny za pomiar i wyświetlanie wyniku na ekranie LCD.

1. Zmienna handler przechowuje natężenie z pinu PC2, pobierane jest 10 wyników pomiaru ADCValue i zapisywane w zmiennej handler,
2. Zmienna handler zapisuje średnią arytmetyczną 10 pomiarów,
3. Wyświetlenie średniego natężenia,
4. Obliczenie średniego napięcia za pomocą wzoru: $(3.3/4095.0) \cdot \text{handler}$ czyli natężenie,
5. Wyświetlenie napięcia.

DMA czyli Direct Memory Access.

DMA służy do szybkiego transferu danych pomiędzy obszarami pamięci oraz pomiędzy peryferiami a pamięcią. Dzięki temu możliwa jest ciągła wymiana danych praktycznie bez udziału procesora.

W celu pominiecia ciągłego wywania funkcji zwracającej wynik pomiaru z ADC, zastosowałem DMA, które odpowiedzialne jest za bezpośrednie przekierowanie danych z peryferii (PC2) do zmiennej ADCValue[]

1. Dodałem nowy kanał DMA i skonfigurowałem go na ADC1
2. Przerwania od DMA włączyłem, a przerwania od ADC wyłączyłem.
3. Uruchomiłem konwersje przetwornika z wykorzystaniem DMA. Utworzyłem zmienną ADCValue, do której wczytałem pomiary. Użyłem w tym celu tablicy jednoelementowej, ponieważ przetwornik będzie wykonywał jeden pomiary w jednej sekwencji konwersji.
4. W funkcji main zadeklarowałem DMAInit.DMA_PeripheralBaseAddr = (**uint32_t**) &ADC1->DR; czyli deklaracja źródła danych, którym jest rejestr ADC1 DR

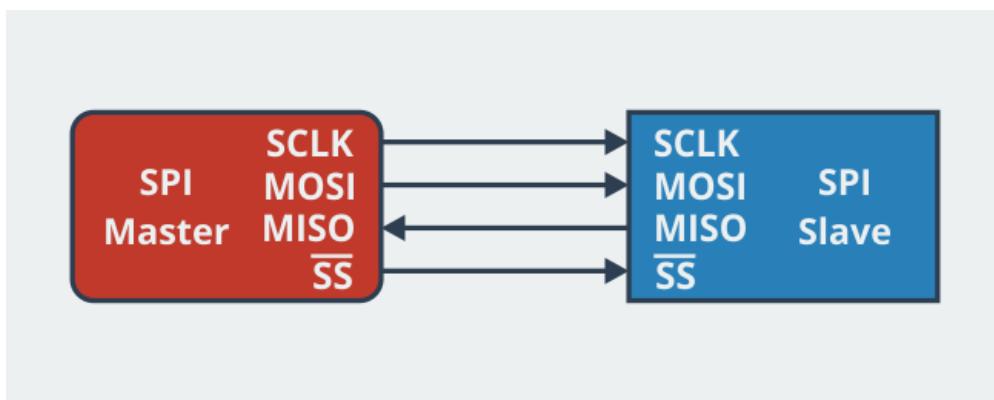
DMA jest bardzo wygodne w użyciu. Po odpowiednim skonfigurowaniu peryferii, wszystko dzieje się automatycznie.

Interfejs SPI

SPI (ang. Serial Peripheral Interface), to szeregowy, synchroniczny interfejs komunikacyjny służący do transmisji danych pomiędzy układami scalonymi. SPI w odróżnieniu od interfejsu I2C do przesyłania danych wykorzystuje 3 linie transmisyjne, co pozwala na komunikację full-duplex.

Zastosowanie poszczególnych linii danych:

- MISO (Master Input Slave Output) – transmisja do urządzenia master.
- MOSI (Master Output Single Output) – transmisja do urządzeń slave.
- SCLK (Serial Clock) – sygnał zegarowy generowany przez urządzenie master.



Transmisja danych przez SPI

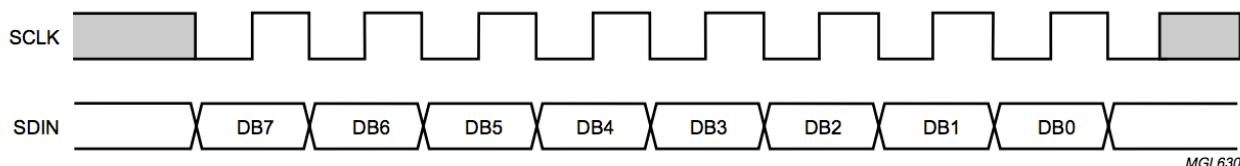
Urządzenie master odpowiada za generowanie sygnału zegarowego na linii SCLK. Dane transmitowane są do podłączonego urządzenia za pomocą linii MOSI. Urządzenie może odpowiadać na linii MISO. Mamy tu więc do czynienia z komunikacją full-duplex.

Przesunięcie w fazie odbierania/wysyłania danych

Drugim z parametrów transmisji, z którym spotkamy się w SPI jest określenie, na którym zboczu zegara następuje odbieranie danych, a na którym transmisja. Odpowiada za to parametr CPHA.

◦ CPHA = 1 – dane są transmitowane na zboczu active⇒idle (przejście ze stanu aktywnego do stanu jałowego). Odbiór danych następuje na zboczu przeciwnym, a więc idle⇒active. Dla parametru CPOL = 0, transmisja danych odbywać się będzie na zboczu opadającym ($3V \Rightarrow 0V$), a odbiór danych na zboczu narastającym ($0V \Rightarrow 3V$).

◦ CPHA = 0 – odwrotnie do sytuacji poprzedniej.



W dokumentacji sterownika PCD8544 można znaleźć rysunek na którym zobaczymy powyższe dane.

Z powyższego rysunku można wywnioskować, że po pierwsze, dane transmitowane są od bitu najstarszego, a więc od MSB (Most Significant Bit).

Po drugie, próbkowanie odbywa się na zboczu narastającym. Oznacza to, że transmisja danych z mikrokontrolera musi się odbywać również na zboczu narastającym.

W przypadku LCD NOKIA 5110 istnieje jednokierunkowy interfejs SPI (nie możemy odczytywać danych z wyświetlacza). Producent zastosował inne oznaczenia linii (CE zamiast CS, DIN zamiast MOSI i CLK zamiast SCLK), jednak sam interfejs działa bez zmian. Wykorzystałem jedynie opcję „Transmit Only Master” z wyłączonym Hardware NSS Signal.

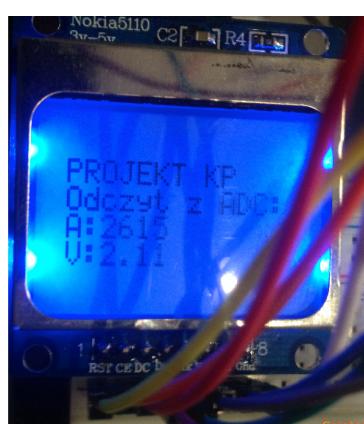
Wyniki działania miernika

Pomiar 1:



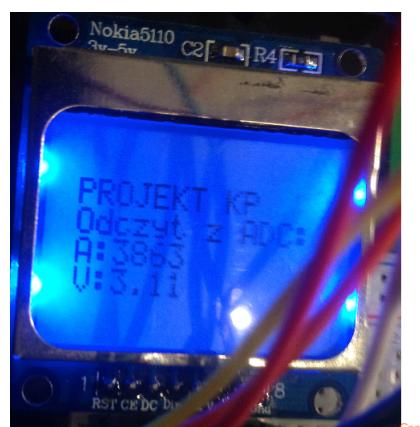
Created by Paint X

Pomiar 2:



Created by Paint X

Pomiar 3:



Created by Paint X