

Sprawozdanie z projektu

„Odbiornik alfabetu Morse’a: czujnik światła jest odbiornikiem, a nadajnikiem impulsy z latarki”

Technika mikroprocesorowa 2

Elektronika i Telekomunikacja, rok III

Michał Szpila

gr. wtorek 9:35

1. Cel projektu

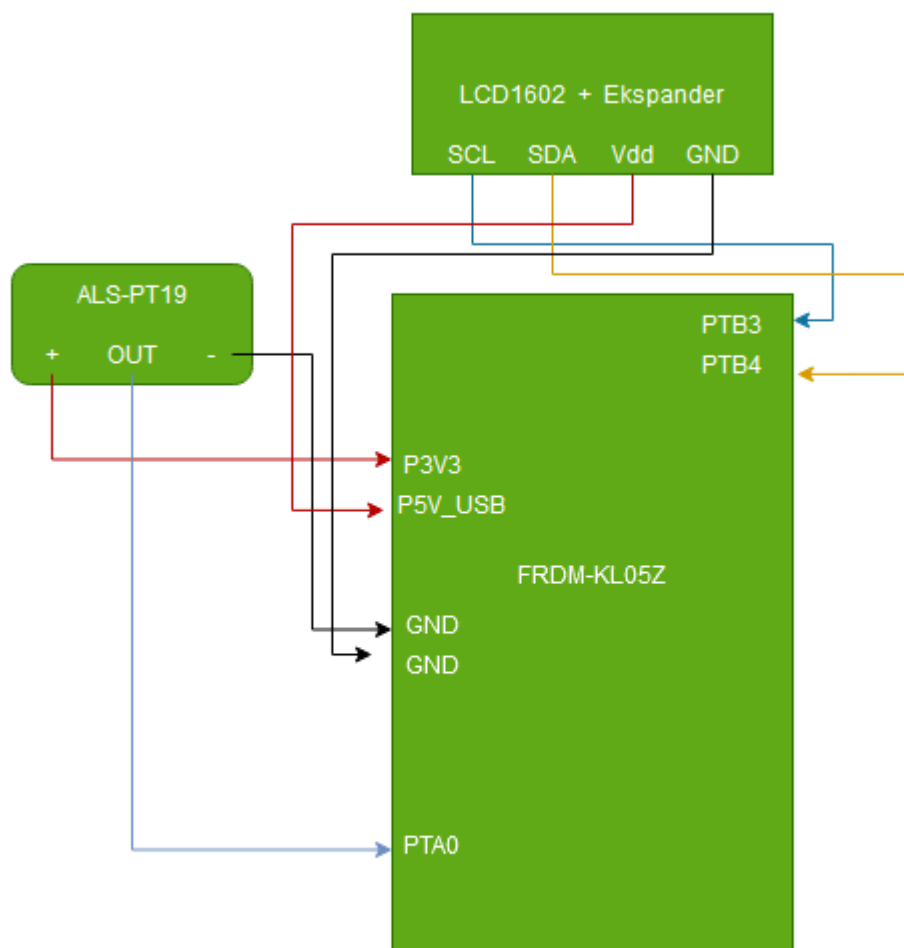
Celem projektu było stworzenie odbiornika alfabetu morse'a, gdzie czujnik światła jest odbiornikiem, a nadajnikiem światło z latarki. Sygnał odebrany przez czujnik światła jest dekodowany i wysyłany przez UART do komputera.

2. Elementy oraz schemat połączeń

Układ zawiera następujące elementy:

- płytke FRDM-KL05Z wraz z wbudowaną diodą RGB,
- czujnik światła ALS-PT19
- wyświetlacz LCD + Ekspander,

Schemat połączeń:



Czujnik światła:

- + → P3V3
- - → GND
- OUT → PTA0

LCD1602 + Ekspander:

- SCL → PTB3
- SDA → PTB4
- Vdd → P5V_USB

3. Działanie programu

Po uruchomieniu odbiornika na drugim wierszu wyświetlacza LCD mogą widnieć trzy komunikaty:

- 1) „Nadawanie w toku”
- 2) „Dekodowanie...”
- 3) „Koniec znaku..”

Pierwszy z nich oznacza że program oczekuje na następny element znaku wysyłanego przez nadajnik. Drugi informuje nas, że program przestał czekać na kolejne elementy znaku i rozpoczyna dekodowanie. Trzeci oznacza, że cały proces dekodowania się zakończył i użytkownik może rozpocząć nadawać kolejny znak. Dodatkowo na wyświetlaczu LCD, możemy po każdym nadanym elemencie zobaczyć jaki ciąg elementów nadaliśmy (np. po nadaniu 3 sygnałów krótkich wyświetlony zostanie „Sygnał: ●●●”). Aby nadawanie stało się łatwiejsze, program informuje nas jak w danym momencie program interpretuje nasz sygnał. Służy do tego dioda RGB, sygnał krótki ‘●’, jest oznaczony kolorem czerwonym, a sygnał długi ‘—’ oznaczony jest kolorem niebieskim. Taki zabieg znacznie utrudnia wysłanie złego sygnału do odbiornika. Aby zapobiec wysłaniu zbyt dużej liczby elementów, przy wysłaniu do odbiornika np. ‘●●—●●—●’ program poinformuje nas o tym na wyświetlaczu LCD oraz zresetuje cały nadany sygnał. Tę funkcjonalność możemy wykorzystać do anulowania nadawanego znaku w przypadku kiedy pomyliłmy element ciągu. Gdy nadany sygnał nie znajdzie odzwierciedlenia w tablicy dekodującej, w terminalu zostanie wyświetlony [0 0].

4. Instrukcja obsługi

Aby zacząć nadawać trzeba tylko zaświecić latarką na czujnik przez odpowiednią ilość czasu. Kolor diody informuje nas w jakim zakresie czasowym w danym momencie jesteśmy. Aby program odczytał element znaku, przestajemy świecić na odbiornik. Na ekranie LCD wyświetla się element ‘●’ lub ‘—’. Następny element możemy nadać kiedy na ekranie pojawi się komunikat „Nadawanie w toku”. Powtarzamy te czynności dotąd osiągniemy pożądany wzór znaku np. ‘—●—’. Po nadaniu wszystkich elementów znaku, czekamy aż pojawi się komunikat „Dekodowanie...” (najpierw pojawi się komunikat „Nadawanie w toku” ale go ignorujemy), który informuje nas że znak właśnie został zdekodowany i wysłany uart’em do komputera. Następnie wyświetlony zostanie komunikat: „Koniec znaku..”, który informuje nas, że możemy rozpocząć nadawać kolejny znak. Nadawać sygnał możemy tylko kiedy komunikat na wyświetlaczu to „Nadawanie w toku” i „Koniec znaku”, ponieważ wtedy program będzie działał poprawnie.

Tabela poniżej przedstawia jak wygląda konwersja sygnału świetlnego na konkretny znak:

•	E	—•	N
•—	A	— — —	O
• — —	W	— — •	G
• — •	R	— • —	K
• — — —	J	— • •	D
• — — •	P	— — — —	^
• — • •	L	— — — •	%
• •	I	— — • —	Q
• • —	U	— — • •	Z
• • •	S	— • — —	Y
• • — •	F	— • — •	C
• • • —	V	• — • —	\
• • • •	H	• • — —	#
—	T	— • • •	B
— —	M	— • • —	X
• — — — —	1	• — • • •	"
• • — — —	2	• • — — • —	_
• • • — —	3	— — — • • •	:
• • • • —	4	— • — • • •	;
• • • • •	5	• • — — • •	?
— • • • •	6	— • — • — —	!
— — • • •	7	— • • • • —	-
— — — • •	8	• — • • •	+
— — — — •	9	— • • • •	/
— — — — —	0	— • — — •	(
• — • — • —	.	— • — — • —)
— — • • — —	,	— • • • —	=
• — — — — •	'	• — — • — •	@

5. Pliki źródłowe

main.c – główny kod programu

lcd1602.c i lcd1602.h – obsługa wyświetlacza lcd

i2c.c i i2c.h - obsługa linii I2C

ADC.h i ADC.c – obsługa przetwornika analogowo-cyfrowego

uart0.c i uart0.h – obsługa portu szeregowego UART