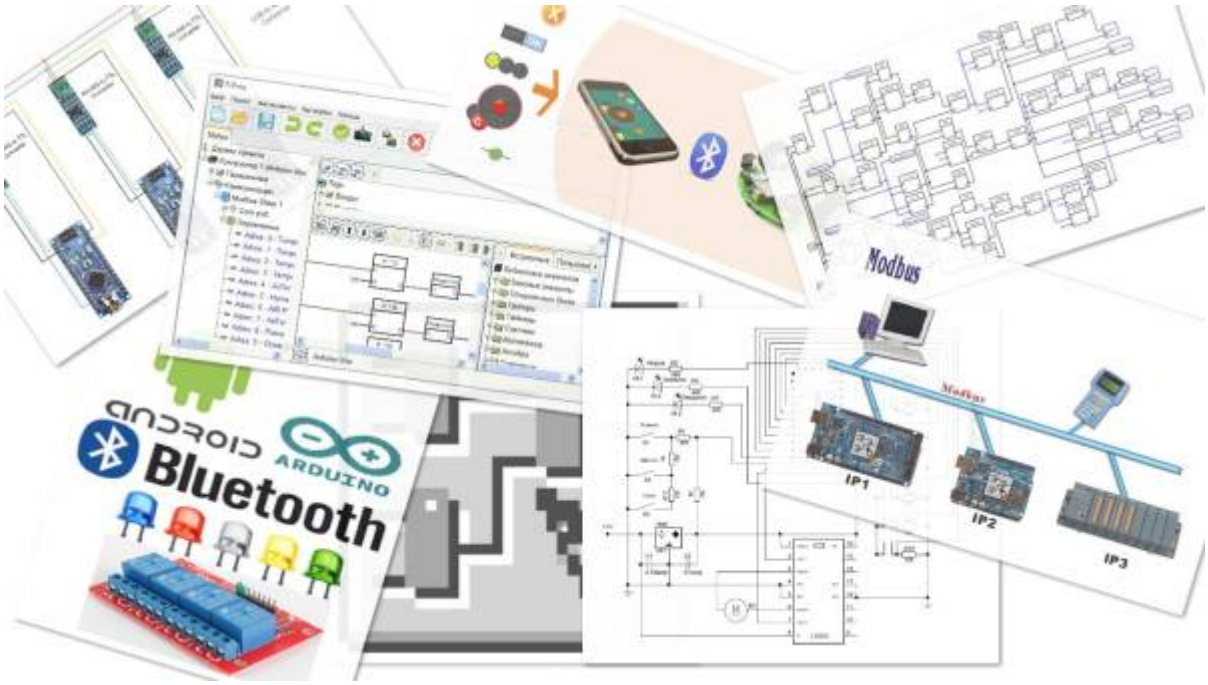
FLprog - Programowanie Arduino dla opornych, Wstęp

oryginalny wpis na blogu: sierpnia 27, 2022



Jest wiele różnych programów pozwalających na programowanie graficzne arduino tj.: ardublock, XODide, MyOpenLab. Natomiast ten program w porównaniu do innych jest zewnętrzny nie jest dodatkiem do Arduino IDE tak jak ardublock. Nie jest zupełnie nowym językiem programowania z bardzo wysokim progiem wejścia, ani również nie wymaga podłączenia do komputera tak jak MyOpenLab. FLprog jest to kompletnie zewnętrzny program który generuje kod który możemy wgrać do arduino. Po zinterpretowaniu naszych graficznych wypocin program po prostu otwiera okno arduino IDE z naszym kodem.

Na stronie autora https://flprog.ru/en/ w zakładce projects możemy zobaczyć wiele rzeczy które społeczność programu zrobiła. Można tam znaleźć projekty sterowników do pieca grzewczego aż po programiki do małych zabawek i robocików.

Program opiera się na bardzo prostej zasadzie mianowicie mamy bloki/klocki z których układamy nasz program, następnie łączymy je przewodami/łącznikami co pozwala na przepływ danych lub stanu logicznego z jednego bloku do drugiego. Pozwala to tak naprawdę na natychmiastową realizację jakiekolwiek pomysłu z praktycznie zerowym progiem wejścia.

Mi udało się do tej pory w tym programie stworzyć coś na wzór czujnika parkowania. Jest to urządzenie które za pomocą czujnika ultra dźwiękowego wydaje periodyczne sygnały dźwiękowe których okresowość jest zależna od dystansu przed czujnikiem ultradźwiękowym. Mówiąc prościej czujnik piszczy częściej jeżeli coś się do niego zbliża.

Poniżej przedstawiam proces tworzenia takiego projektu.

Najpierw musimy znaleźć lub zakupić części, w tym projekcie wykorzystano poniższe części:

- Czujnik zbliżeniowy HCSR04
- Mikrokontroller Arduino UNO

FLprog wspiera Arduino Uno, Arduino Leonardo, Arduino MEGA i wiele innych

- Moduł Wzmacniacza Audio 18W TDA2030 18W

Buzzery do arduino mają bardzo drażniący dzwięk to pozwoli na lekkie zniwelowanie tego efektu

- Jakiś Stary głośnik

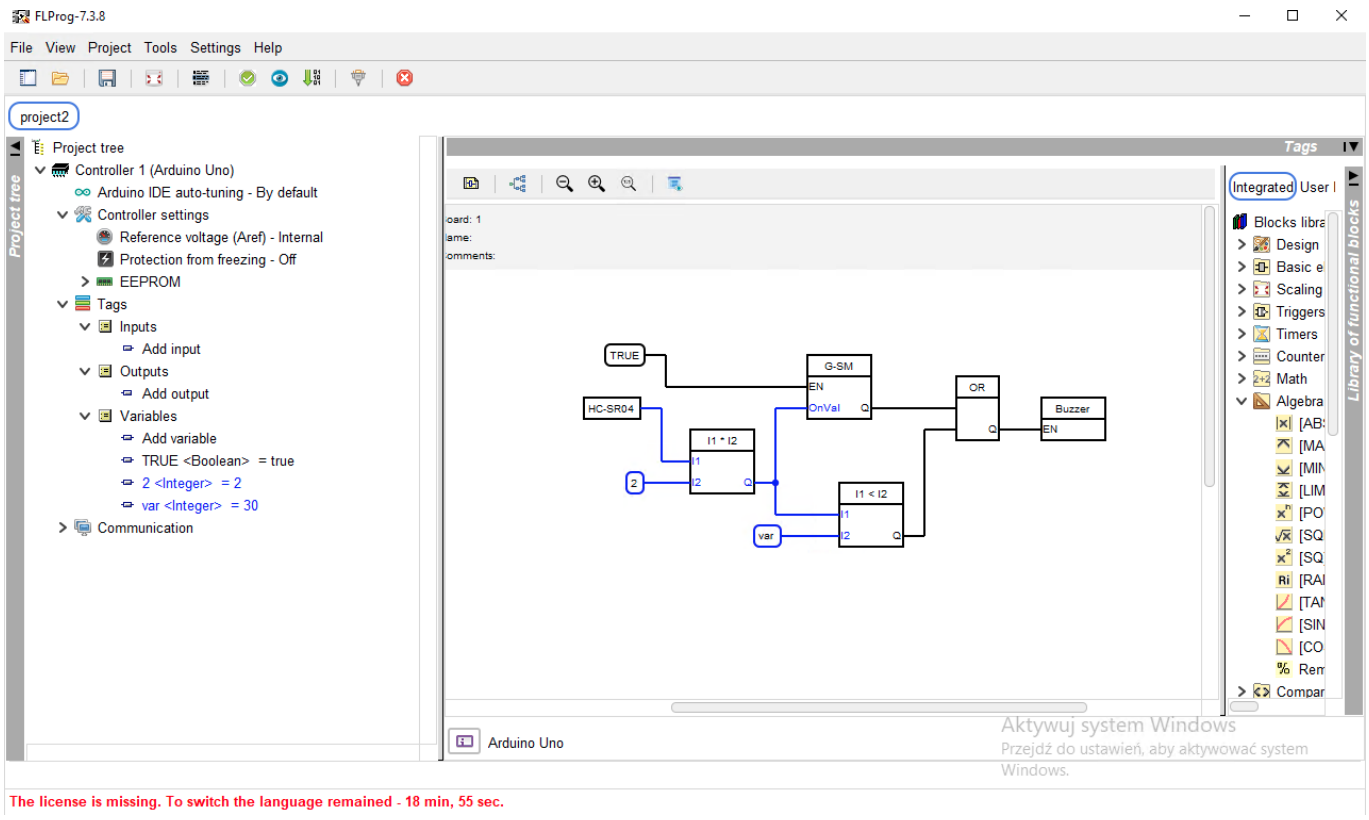
impedancji od 3 ohm do 16 ohm

- Zródło zasilania

W moim projekcie arduino jest zasilane z zasilacza warsztatowego Natomiast wzmacniacz audio jest zasilany z linii 5V z arduino gdyż nic tutaj nam nie będzie pobierać zawrotnych wartości prądu

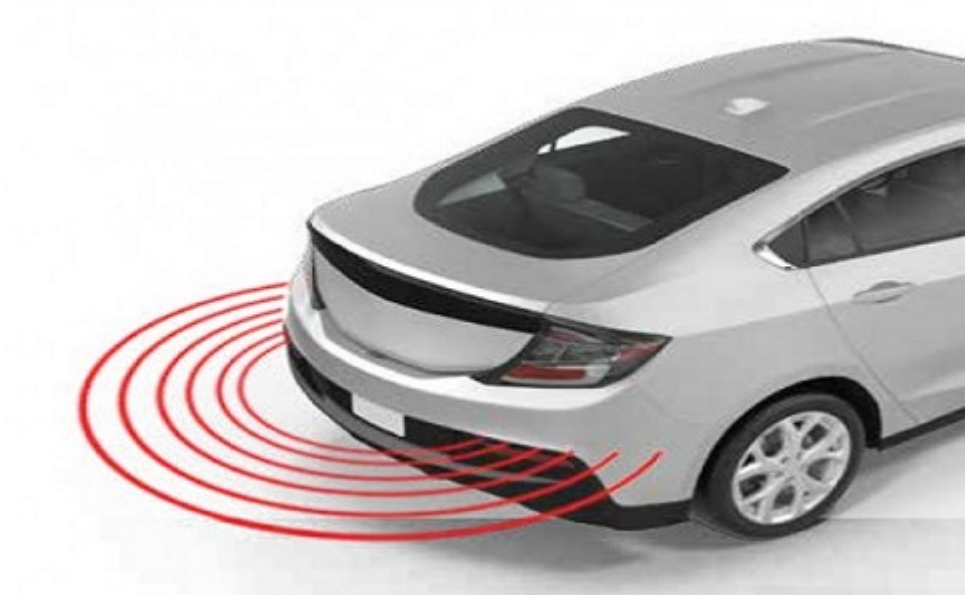
Potem należy wszystko podłączyć. Czujnik Podłączamy do pinów arduino nie jest istotne które konkretnie ważne aby miały znaczek ~ bo to oznacza że wpierają PWM a to jest dla nas istotne, w programie potem można skorygować numer wykorzystanego pinu. Następnie podłączamy wzmacniacz, podłączamy jego zasilanie podłączamy głośnik oraz na wejście wzmacniacza podłączmy jeden z pinów.

Następnie otwieramy program i zaczynamy zabawę. Poniżej mamy już cały wykonany projekt. Przy tworzeniu projektu naciskamy File>New>New Project for the controller a następnie w dodatkowym okienku wybieramy opcję FBD (function block diagram) czyli właśnię tą opcję o którą nam chodzi. Z listy wybieramy mikrokontroller który nas interesuję.



W ten sposób stworzyliśmy nowy pusty projekt.

Teraz warto wyjaśnić co mamy na ekranie bo nie jest tego mało. Po prawej stronie interfejsu mamy dostępny bloki które pozwalają nam tworzyć program. Po lewej stronie mamy widoczne nasze: zmienne, wejścia i wyjścia arduino, porty komunikacyjne, pamięć wewnętrzną oraz zabezpieczenie przez zacięciem mikrokontrollera. Następnie żeby zacząć prace normalnie musieli byśmy dodać wejścia i wyjścia natomiast w tym przypadku będziemy korzystać z gotowych bloków przystosowanych do naszych elementów. Zaczynamy od przeciągnięcia z listy z prawej strony bloku naszego czujnika HC-SR04 zakładka sensors>distance sensor. Klikając dwa razy w te blok możemy ustawić na których pinach podłączyliśmy sygnały Echo i Trig naszego sensora. W nocie katalogowej możemy wyczytać że sensor ten maksymalnie jest w stanie wykryć obiekty oddalone o 4. W tym oknie ustawiamy również ten dystans. Teraz mamy blok funkcyjny który na wyjściu podaje nam w formacie integer wartość odległości obiektu od sensora. FLprog podświetla nam takie połączenia jak i też zmienne integer na niebiesko. Teraz z racji tego że nasza liczba z sensora raczej będzie mała ponieważ interesują nas obiekty blisko, z zamysłem że budujemy czujnik prakowania. W związku z tym tworzymy zmienną integer o wartości 2 oraz pobieramy blok mnożenia math>MUL(*). Łączymy wszystko tak jak na obrazku, mnożenie jest przemienne więc nie ważne do którego wejścia bloku podłączymy co. Następnie chcemy generować impulsy od długości proporcjonalnej do odległości. Pobieramy z zasobnika blok generatora Timers>Generator. Następnie w parametrach tego bloku ustawiamy nasz generator jako symetryczny multiwibrator a następnie wybieramy że długośc impulsów jest zależna od zewnętrznego parametru. Generator musimy również włączyć więc tworzymy zmienną boolean o wartości true i łączymy ją do wejścia enable. Następnie musimy stworzyć warunek że poniżej pewnej odległości głośnik zacznie piszczeć w sposób ciągły. Wykorzystujemy do tego blok comparison>comparator. I wybieramy odpowiednie porównanie tworzymy zmienną dla minimalnej odległości i łączymy wszystko w odpowiedni sposób i teraz nasz komparator będzie miał wyjście w stanie HIGH wtedy kiedy odległość spadnie poniżej około 30cm. Następnie musimy w jakiś sposób połączyć te dwa sygnał w taki sposób że każdy z nich jest równoważny czyli jak potrzeba to generator włącza głośnik. Osoby które znają bramki logiczne powiedzą … tutaj pasuje OR i tak to prawda właśnie o taką bramkę chodzi, podłączamy ją. I teraz zostaje ostatni blok other>Piezo Speaker, dodajemy go do projektu podłączamy do wyjścia bramki OR i ustawiamy parametry, w ustawieniach ustawiamy pin który podłączyliśmy do wejścia wzmacniacza, wybieramy również opcję continuosly opcja jest wyjaśniona w okienku, a następnie częstotliwość jaką chcemy żeby generował, ja ustawiłem na 1kHz uważam że jest stosunkowo mało irytujący taki dzwięk.



W ten sposób mamy czujnik parkowania co prawda bardzo prosty natomiast działa. Zapraszam do sprawdzenia innych moich projektów oraz polecam samemu zacząć przygodę z arduino i FLprog.