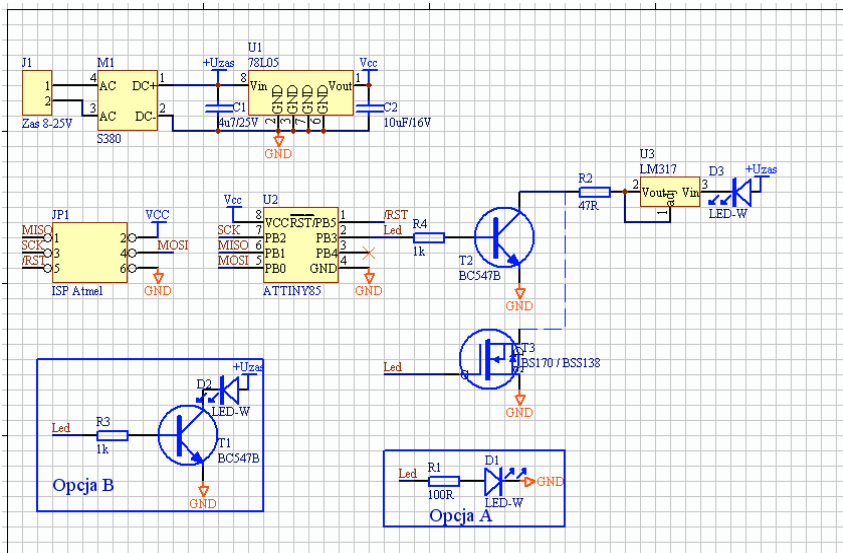


Symulator efektu spawania do makiety

Przeszukując Internet napotkałem nagranie video przedstawiające efekt spawania na makiecie. Spodobało mi się to i poszukałem sposobów generowania takiego efektu. Niestety znalazłem tylko rozwiązania oparte o kilka połączonych ze sobą generatorów (np. NE555). Niestety przypadkowość efektu jest dość słaba. Przedstawiony tu projekt wykorzystuje generator pseudolosowy procesora. Możliwe jest ustawienie kilku parametrów, dzięki czemu efekt można dopasować do własnych potrzeb (np. stworzyć efekt wyładowań atmosferycznych).



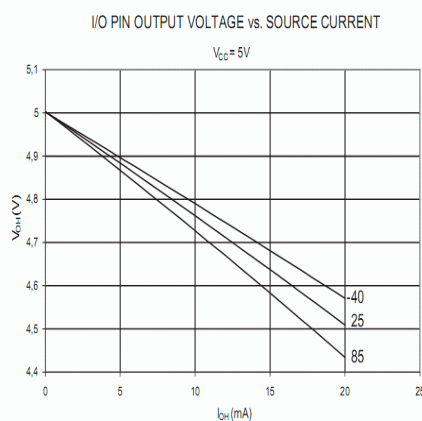
Budowa i zasada działania:

Budowa urządzenia jest bardzo prosta. Napięcie zasilania jest prostowane w mostku M1 oraz stabilizowane w stabilizatorze U1. Funkcje sterującą pełni mikrokontroler rodziny AVR Attiny85. Program w nim umieszczony generuje przebieg pseudolosowy na wyprowadzeniu PB3 (nóżka 2). Mikrokontroler może bezpośrednio sterować białą „zimną” diodą LED imitującą światło spawarki (na schemacie „Opcja A”). Trzeba jednak mieć na uwadze pewne ograniczenia. Maksymalna wydajność prądowa portu procesora w stanie wysokim wynosi 20mA. Ogranicza to więc prąd diody do takiej właśnie wartości. Aby obliczyć wartość rezystora R1 należy posłużyć się prawem Ohma. Jednak znany wszystkim wzór:

$$R = U / I$$

trzeba trochę zmodyfikować ponieważ w obliczeniach należy jeszcze uwzględnić spadek napięcia na U2, który dla prądu 20mA wyniesie około 0,5V (wykres z noty katalogowej układu ATtiny85)

Figure 22-22. I/O Pin Output Voltage vs. Source Current ($V_{CC} = 5V$)



oraz spadek napięcia na diodzie led (w przybliżeniu 3..3,6V zależnie od typu diody). Wzór będzie więc miał postać:

$$U = (U_{zas} - U_{led} - U_{cpu}) / I$$

$$U = (5 - 3 - 0,5) / I$$

$$U = 1,5 / I$$

Wartość rezystora R1 nie może więc być mniejsza niż 75R, bo został by przekroczony prąd 20mA, co mogło by uszkodzić mikrokontroler. Fakt, że dioda świeci impulsowo i mikrokontroler w impulsie wytrzyma większy prąd nie rozwiązuje problemu. Może się bowiem zdarzyć, że z powodu błędu w programie dioda zaświeci się na stałe, a tego mikrokontroler może już nie wytrzymać. Ponadto, przy takim rozwiązaniu trzeba mieć też na uwadze moc wydzielaną w stabilizatorze U1. Przy wysokim napięciu zasilającym U1, moc na nim wydzielona może wynieść:

$$P = (U_{zas} - U_{stab}) * I$$

$$P = (25 - 5) * 0,02A$$

$$P = 0,4W.$$

Przy takiej mocy nie można już zastosować miniaturowego stabilizatora 78L05, lecz 7805 w obudowie TO-220. Taka obudowa bez problemu poradzi sobie z mocą 1W bez radatora.

Jakie więc zmiany należy wprowadzić aby osiągnąć większy prąd? Pierwsze rozwiązanie jakie się nasuwa to dołożenie tranzystora i z niego sterowanie diody. Ze względu na to, że napięcie zasilania 5V jest znacznie zmniejszone przez spadek napięcia na diodzie led, diodę należało by podłączyć do napięcia niestabilizowanego, zaraz za mostkiem prostowniczym (opcja B). Problemem może być jednak osiągnięcie granicznego prądu diody, w przypadku gdy napięcie zasilające nie jest zbyt stabilne. W takiej sytuacji prąd może wzrosnąć z ponad dopuszczalny prąd diody, zmniejszając jej żywotność. Rozwiązanie takie było by dobre tylko w przypadku stabilnego napięcia zasilającego urządzenie lub prądu diody znacznie odbiegającego od granicznego.

Aby rozwiązać problem stabilności napięcia zasilającego należy zastosować źródło prądowe. Źródło takie można zbudować z diody Zenera, tranzystora i rezystora, ale prościej zastosować układ LM317. Prąd źródła, czyli także włączonej szeregowo z nim diody, wyniesie:

$$I = 1.25 / R2$$

Dla rezystora R2 o wartości 47R otrzymamy prąd 26mA, dla 22R – 56mA, dla 10R – 125mA.

Wzór ten będzie jednak prawdziwy dla napięcia zasilającego wynoszącego minimum:

$$U = U(lm317) + U(led D3) + U(ce T2) + U (M1)$$

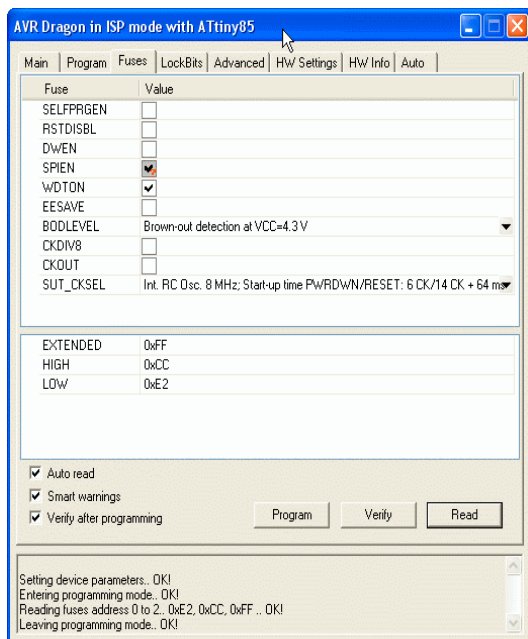
$$U = 1.25 + 3.6 + 0,2 + 1,2$$

$$U = 6,25V$$

W praktyce napięcie zasilające musi być wyższe, chociażby z powodu stabilizatora U1. Co prawda układy 78Lxx do poprawnej pracy wymaga napięcia zasilającego wyższego o 100mV, należy jednak uwzględnić dodatkowy spadek napięcia na diodach mostka M1.

Montaż i uruchomienie:

Montaż elementów jest tradycyjny i nie wymaga omawiania. Jeśli nie przewiduje się programowania procesora U2 w systemie, to można nie montować złącza JP1. Zamiast tranzystora bipolarnego T2, można zastosować mosfet np. BS170. W takim przypadku rezystor R3 jest zbędny i może zostać zastąpiony zworą. Mikrokontroler należy zapisać programem: „spawarka.hex” lub „burza.hex”. Ustawienie bitów konfiguracyjnych przedstawia rysunek:



Filmy z efektem wygenerowanym przez w/w opisane urządzenie można obejrzeć na:

<http://www.youtube.com/watch?v=rOYZmbyrT58>

<http://www.youtube.com/watch?v=EgbQo8ldX94>

<http://www.youtube.com/watch?v=AH4pTj55bFU>

Dla zaawansowanych:

Przedstawiony projekt nie jest ostatecznym rozwiązaniem i można go rozwijać. Program można zmodyfikować tak, aby generował kilka przebiegów pseudolosowych na różnych wyprowadzeniach. Kody źródłowe są dostępne na stronie EP oraz na <http://www.kolejki.eu>. Jeśli chcemy zmodyfikować czasy odpowiedzialne za generowanie efektu to zależy zmodyfikować następujące linie kodu źródłowego:

```
#define PracaMin 60 // sek
#define PracaMax 180
#define PrzerwaMin 120
#define PrzerwaMax 240

#define SpawaMin 1000 // ms
#define SpawaMax 5000
#define NieSpawaMin 2000
#define NieSpawaMax 9000

#define BlyskMin 5 // ms
#define BlyskMax 100
```

PracaMin/Max decyduje o zakresie (ustalonym losowo ze zdefiniowanego zakresu) czasu pracy spawacza. PrzerwaMin/Max decyduje o czasie przerwy w pracy. SpawaMin/Max decyduje jak długo trwa wykonanie jednego spawu (czas serii błysków) w czasie pracy spawacza, NieSpawaMin/Max ustala przerwy pomiędzy kolejnym spawem (serią błysków). BlyskMin/Max decyduje o czasie trwania pojedynczego błysku.

Dla symulacji efektu wyładowań atmosferycznych czasy te wynosiły by:

```
#define PracaMin 2 // sek
#define PracaMax 4
```

```
#define PrzerwaMin 30
#define PrzerwaMax 240

#define SpawaMin 50 // ms
#define SpawaMax 200
#define NieSpawaMin 5000
#define NieSpawaMax 10000

#define BlyskMin 5 // ms
#define BlyskMax 20
```

Sławomir Skrzyński, EP
slawomir.skrzynski@ep.com.pl