

Inkrementalny pomiar kąta (019; 17.09.2009; arduino)

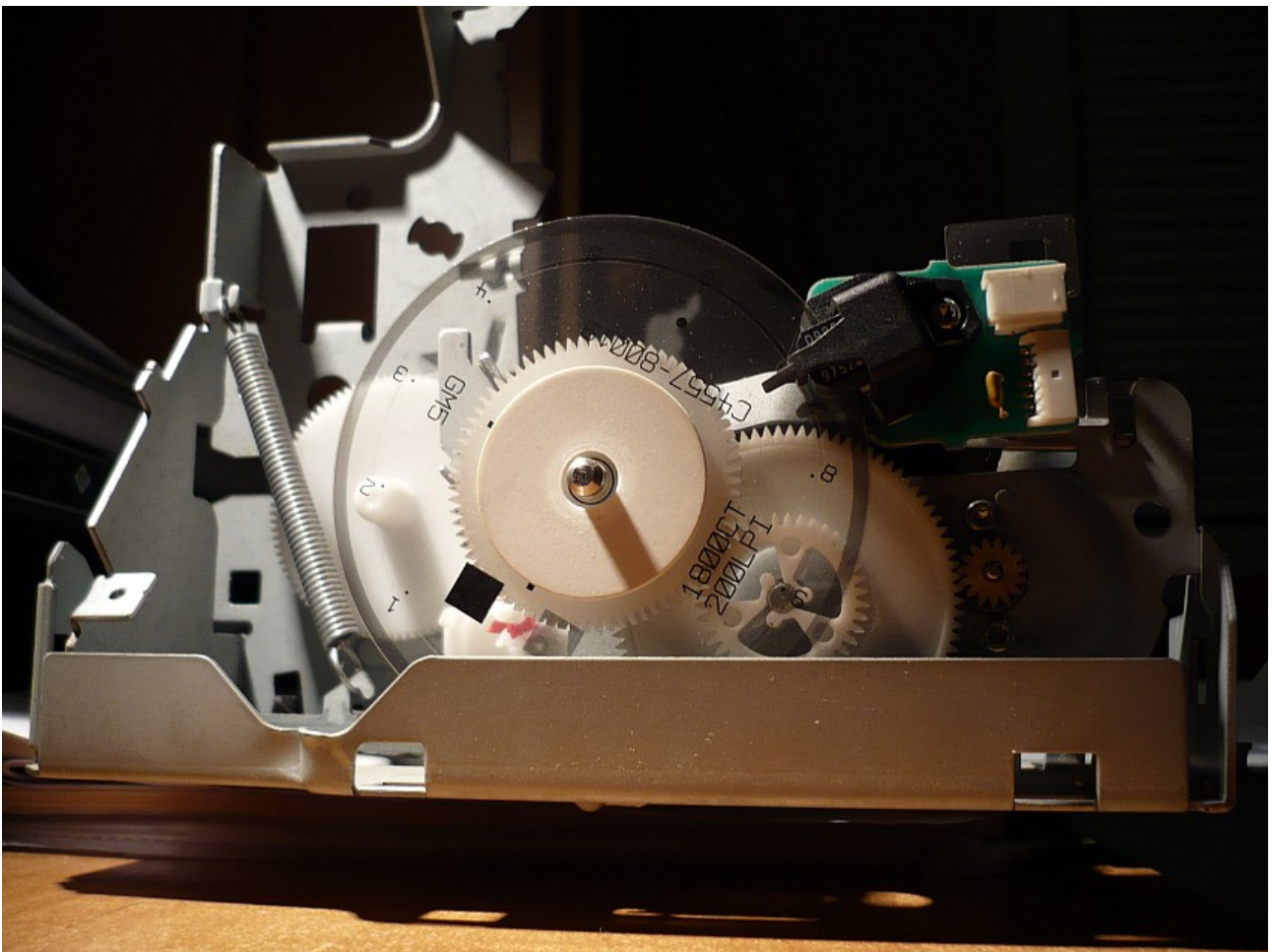
Opis metody pomiaru kąta obrotu z użyciem inkrementalnego czujnika optycznego oraz Arduino.

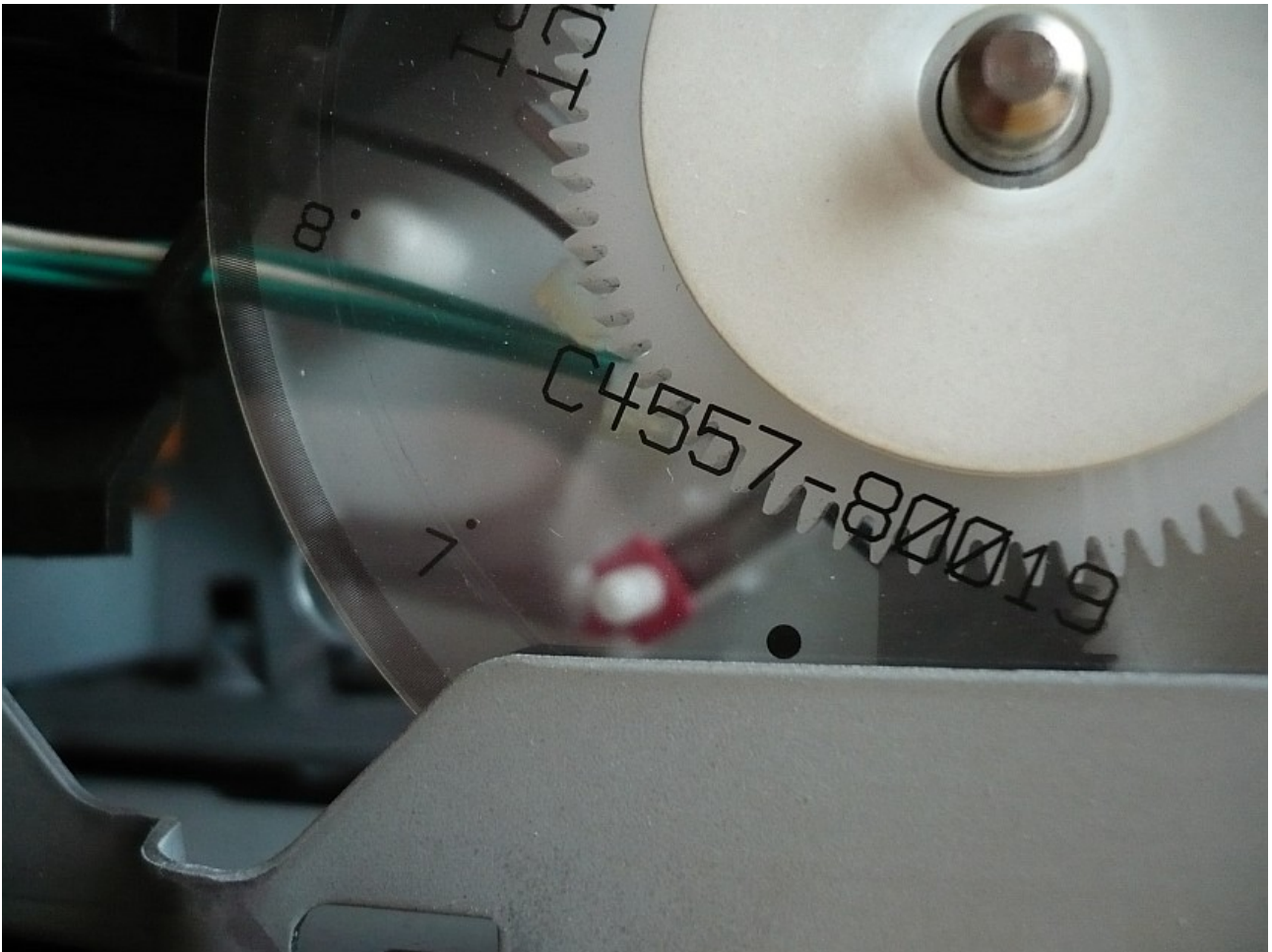
Jednym ze sposobów pomiaru przemieszczenia kąowego jest zastosowanie inkrementalnego czujnika optycznego. Do pełnego pomiaru niezbędne będą zatem 3 podstawowe elementy:

- wzorzec inkrementalny,
- czujnik inkrementalny,
- enkoder inkrementalny.

Wzorzec inkrementalny.

Jest to przezroczysty krążek z naniesionymi w stałych odległościach liniami. Może wyglądać następująco:





Pokazany wzorec wymontowany został z napędu przesuwu papieru starej drukarki atramentowej.

Charakteryzują go dwie wartości:

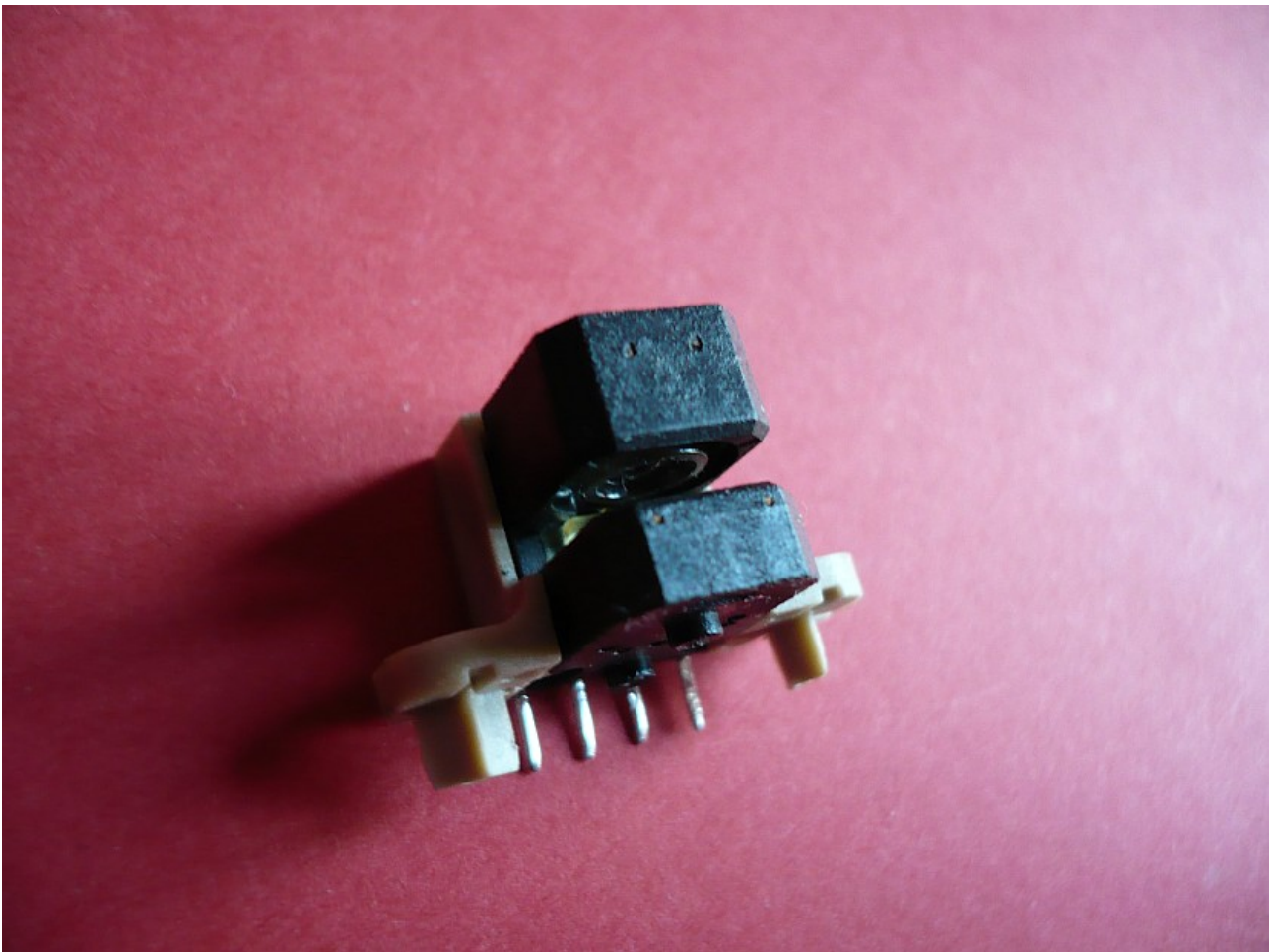
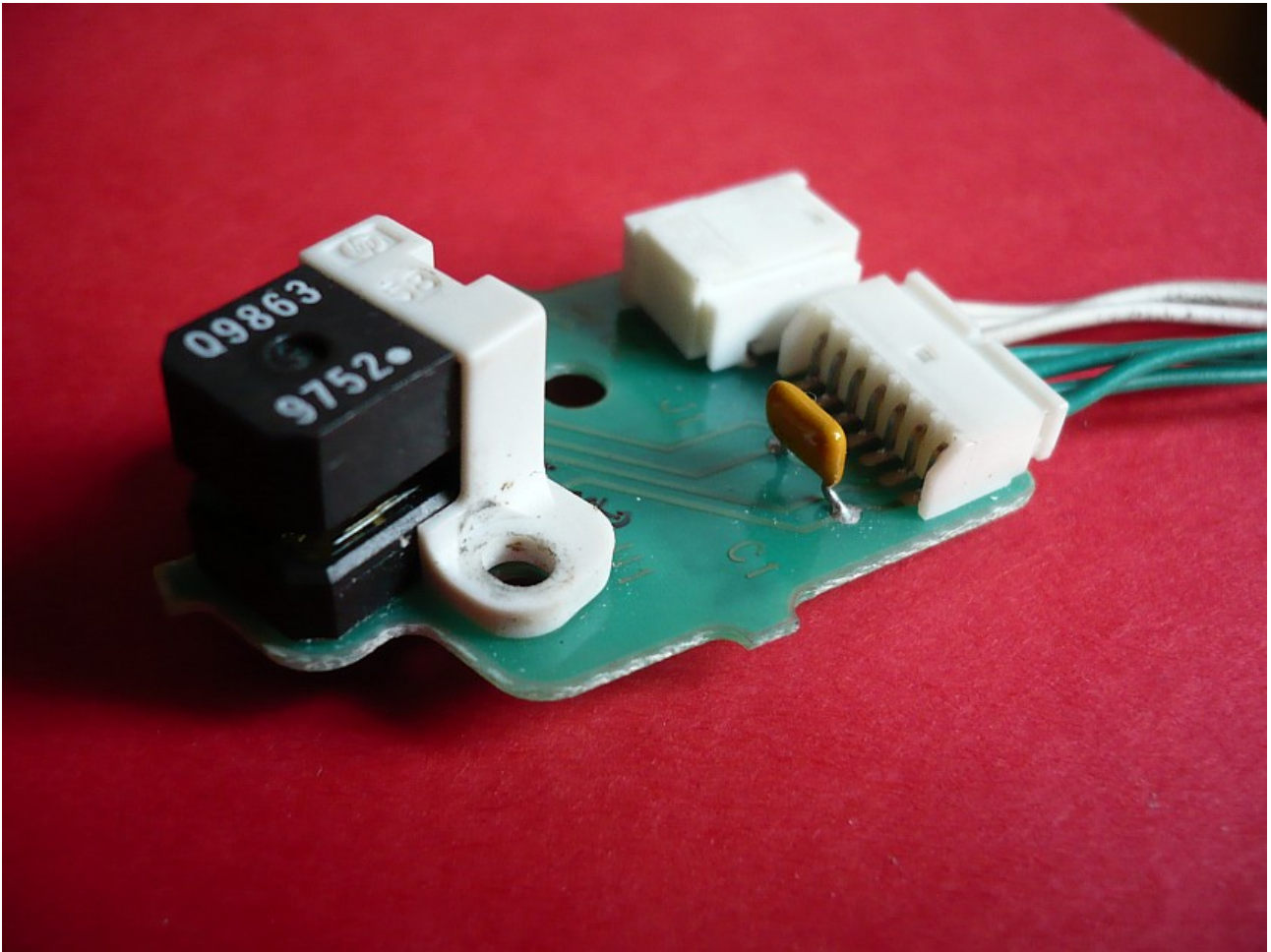
1800CT – oznacza liczbę linii przypadających na cały obwód tarczy,

200LPI – oznacza liczbę linii przypadających na cal obwodu tarczy.

Czujnik inkrementalny.

Element elektroniczny pozwalający wykrywać zmiany przezroczystości tarczy inkrementalnej. Standardowo składa się on z diody i odbiornika światła (fotorezystor lub fototranzystor). Często czujnik ma wspólną obudowę i posiada dwa odbiorniki światła nieznacznie względem siebie przesunięte (co pozwala wykrywać kierunek przemieszczania wzorca inkrementalnego).

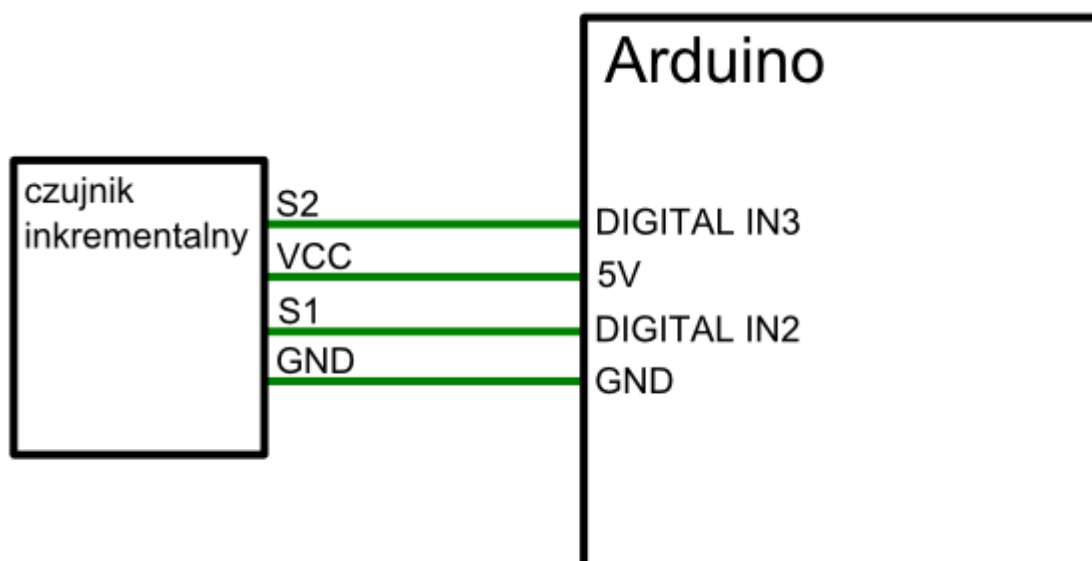
W przykładzie zastosuję czujnik inkrementalny użyty w w/w drukarce, jak na zdjęciu:



Enkoder inkrementalny.

Enkoder inkrementalny przetwarza impulsowy sygnał odczytany z czujnika inkrementalnego na wartość kąta. W naszym przypadku rolę tą spełniać będzie Arduino i specjalnie napisany do tego program.

Podłączenie czujnika inkrementalnego realizujemy jak na poniższym schemacie:

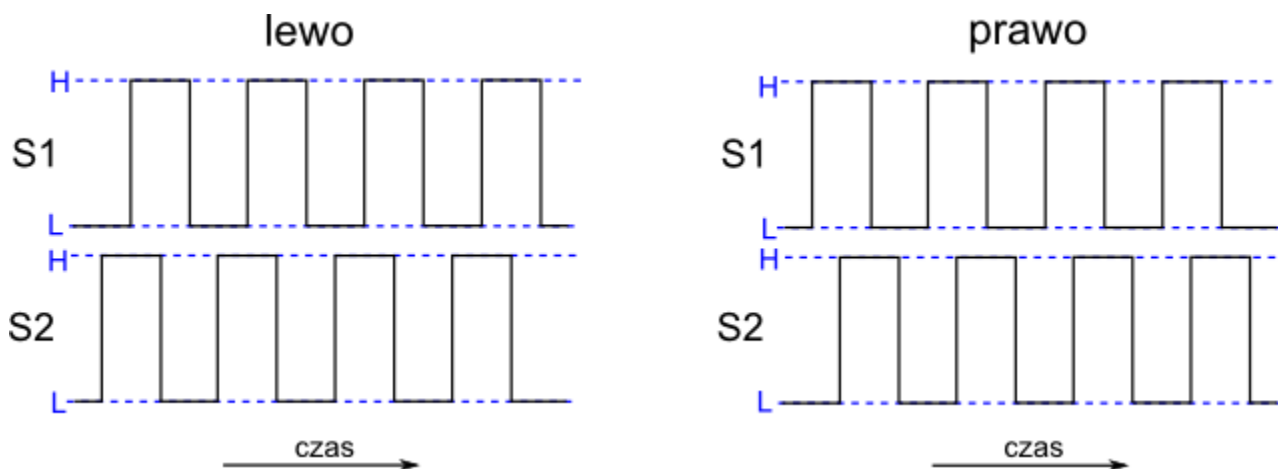


Idea pomiaru.

Pomiar kąta polegać będzie na rozpoznawaniu przez Arduino kolejnych zmian sygnałów otrzymywanych z czujnika inkrementalnego. Niezbędne będzie do tego zastosowanie tzn. przerw (ang. interrupt). W mikroprocesorze Atmega 328 dostępne mamy dwa piny mogące wykrywać przerwy, czyli zmiany stanu sygnału z poziomu wysokiego na niski lub odwrotnie. W Arduino Duemilanove są to wejścia cyfrowe numer 2 i 3.

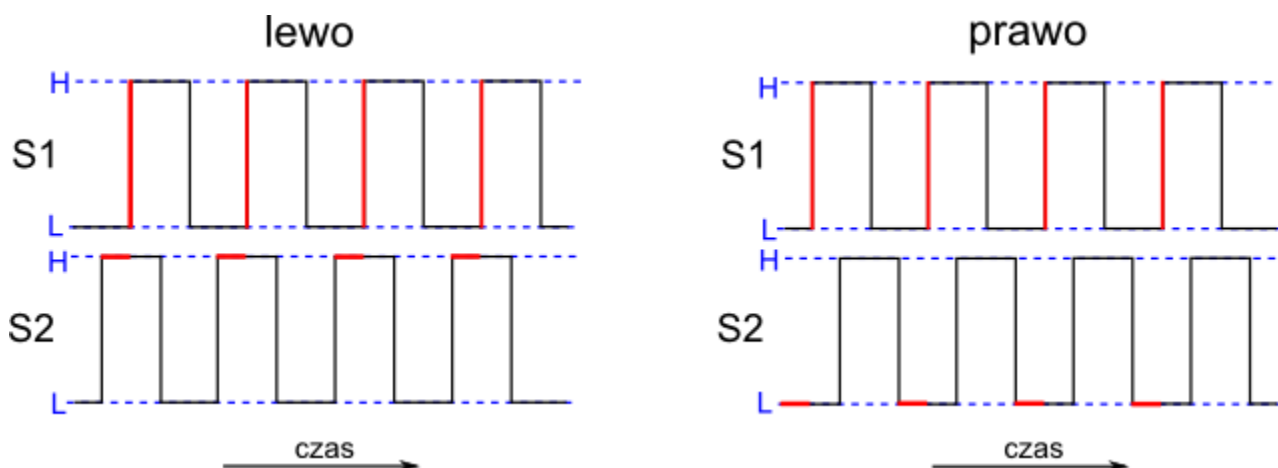
Jak zatem wykorzystać przerwy i mając dwa sygnały z czujnika przesunięte względem siebie rozpoznać kierunek obrotu?

Założmy obracanie się tarczy inkrementalnej w lewo lub w prawo. Przebieg zmian sygnałów z czujnika w czasie wyglądać będzie następująco:



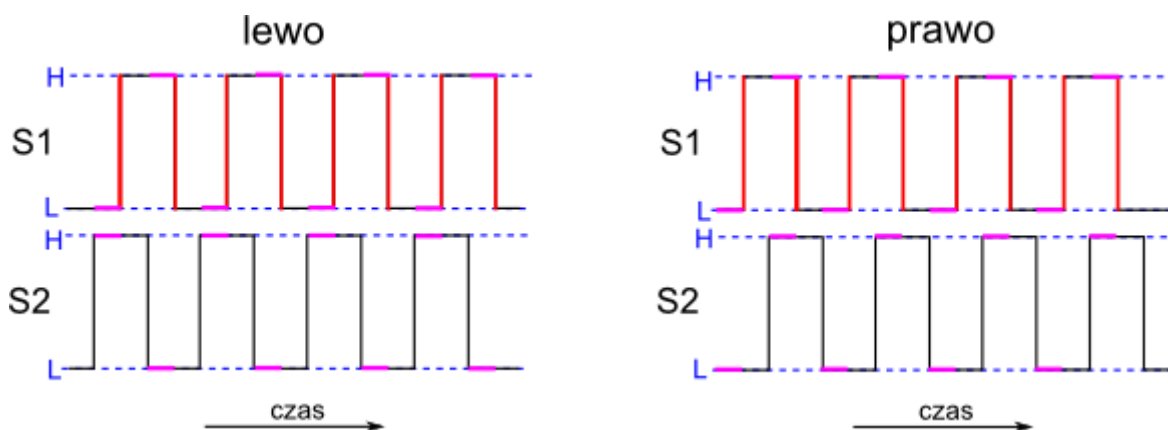
Założmy, że korzystamy z przerwania, które wykrywa tylko zbocze narastające sygnału S1 i nie dokonujemy pomiarów sygnału S2. Otrzymujemy zatem dokładność pomiaru kąta obrotu równą 360 stopni podzielone przez liczbę linii na obwodzie wzorca inkrementalnego (w naszym przypadku $360^\circ/1800=0,2^\circ$). Jednocześnie możemy dokonywać pomiarów tylko przy obrocie w jednym kierunku (jedna zmiana kierunku spowoduje już błąd pomiaru).

Założmy więc, że korzystamy z przerwania, które wykrywa tylko zbocze narastające sygnału S1 i jednocześnie możemy dokonywać pomiaru stanu sygnału S2. Zwróćmy uwagę, jaką wartość ma sygnał S2 przed i w czasie wystąpienia narastającego zbocza sygnału S1 dla obrotu w jednym bądź drugim kierunku:



Widzimy zatem, że pomiar stanu logicznego sygnału S2 tuż przed wystąpieniem przerwania na sygnale S1 pozwala rozpoznać kierunek obrotu. Oczywiście algorytm może badać stan S2 również tuż po wystąpieniu przerwania.

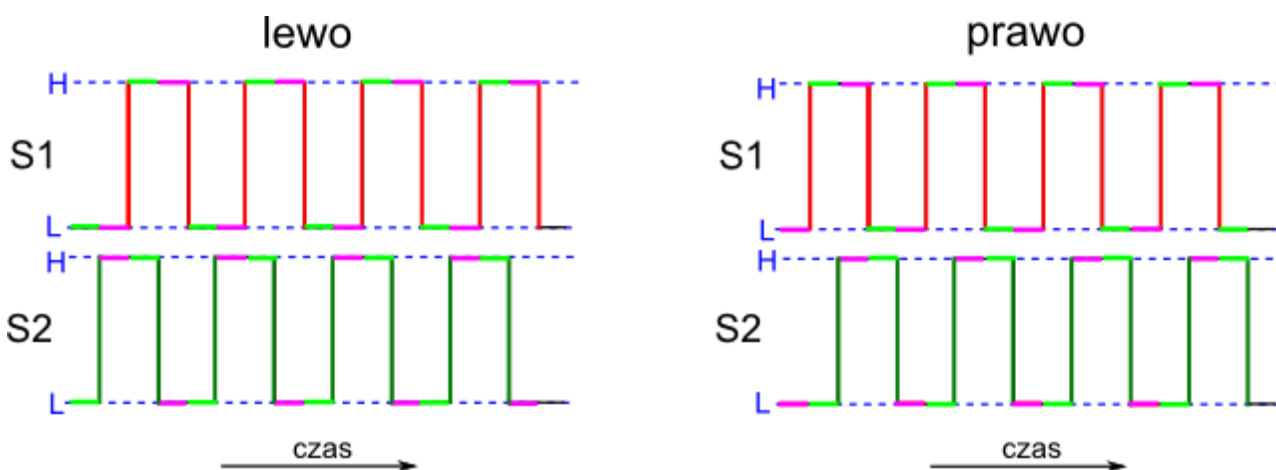
Co stanie się, gdy na sygnale S1 zastosujemy przerwanie wykrywające zarówno narastające jak i opadające zbocze?



Enkoder będzie miał teraz rozdzielczość dwukrotnie większą niż w pierwszym przypadku ($360^{\circ}/3600=0,1^{\circ}$). Traktując obrót w lewo jako wzrost wartości kąta, należy:

- zwiększyć kąt o $0,10^{\circ}$ w przypadku gdy różnią się stany S1 i S2 poprzedzające wykrycie zmiany stanu S1,
- zmniejszyć kąt o $0,10^{\circ}$ w przypadku gdy takie same są stany S1 i S2 poprzedzające wykrycie zmiany stanu S1.

Największe możliwości daje jednak zastosowanie dwóch przerwań dla sygnałów S1 i S2 z możliwością sprawdzania ich stanów logicznych. Widać to na poniższej ilustracji:



Enkoder będzie miał teraz rozdzielczość aż czterokrotnie większą niż w pierwszym przypadku ($360^{\circ}/7200=0,05^{\circ}$).

Kąt należy zwiększyć o $0,5^{\circ}$ w przypadku, gdy przerwanie sygnału S1 poprzedzają różne poziomy S1 i S2, lub gdy przerwanie sygnału S2 poprzedzają takie same poziomy S1 i S2.

Kąt należy zmniejszyć o $0,5^{\circ}$ w przypadku, gdy przerwanie sygnału S1 poprzedzają takie same poziomy S1 i S2, lub gdy przerwanie sygnału S2 poprzedzają różne poziomy S1 i S2.

Arduino jako enkoder inkrementalny.

Kod do Arduino stosujący ostatnią metodę, może wyglądać następująco:

```
volatile boolean a = LOW;
volatile boolean b = LOW;
volatile int alfa = 0;
void setup(){
  Serial.begin(115200);
  pinMode(2, INPUT);
  pinMode(3, INPUT);
  attachInterrupt(0, A, CHANGE);
  attachInterrupt(1, B, CHANGE);
}
void loop(){
  a=digitalRead(2);
  b=digitalRead(3);
  Serial.println(alfa/20);
  delay(100);
}
void A(){
  if(b==a)
  {
    alfa--;
  }
  else
  {
    alfa++;
  }
  a=digitalRead(2);
  b=digitalRead(3);
}
void B(){
  if(b==a)
  {
    alfa++;
  }
  else
  {
    alfa--;
  }
  a=digitalRead(2);
  b=digitalRead(3);
}
```

Inicjacja zmiennej programu poprzedzona została komendą *volatile*, która sprawia, że kompilator nakazuje mikroprocesorowi przechowywać zmienną w pamięci RAM zamiast w rejestrze. Taka kwalifikacja zmiennej będzie właściwie niezbędna tylko w przypadku pracy z przerwaniami.

`attachInterrupt(0, A, CHANGE)` – komenda uruchamiająca wykrywanie przerw w mikrokontrolerze na porcie nr 0 (odpowiada pinowi wyjścia/wejścia cyfrowego nr 2 Arduino Duemilanove). Gdy wykryta zostanie zmiana poziomu logicznego sygnału bezwzględnie wykonany zostanie funkcja A. Po wykonaniu funkcji kontynuowane będą zadania pętli głównej programu `loop()`.

Należy zwrócić uwagę, że prędkość zmiany sygnału odtrzymywanego z czujnika ograniczona jest możliwościami wykonywania obliczeń przez Arduino (funkcja jednego przerwania musi wykonać się zanim nastąpi drugie przerwanie). Opisaną metodę możemy z powodzeniem zastosować do pomiaru prędkości, ale z użyciem wzorca inkrementalnego o wielokrotnie mniejszej liczbie linii (np 4 linie na obwód, przy czym szerokości linii i pól wolnych powinny być takie same).