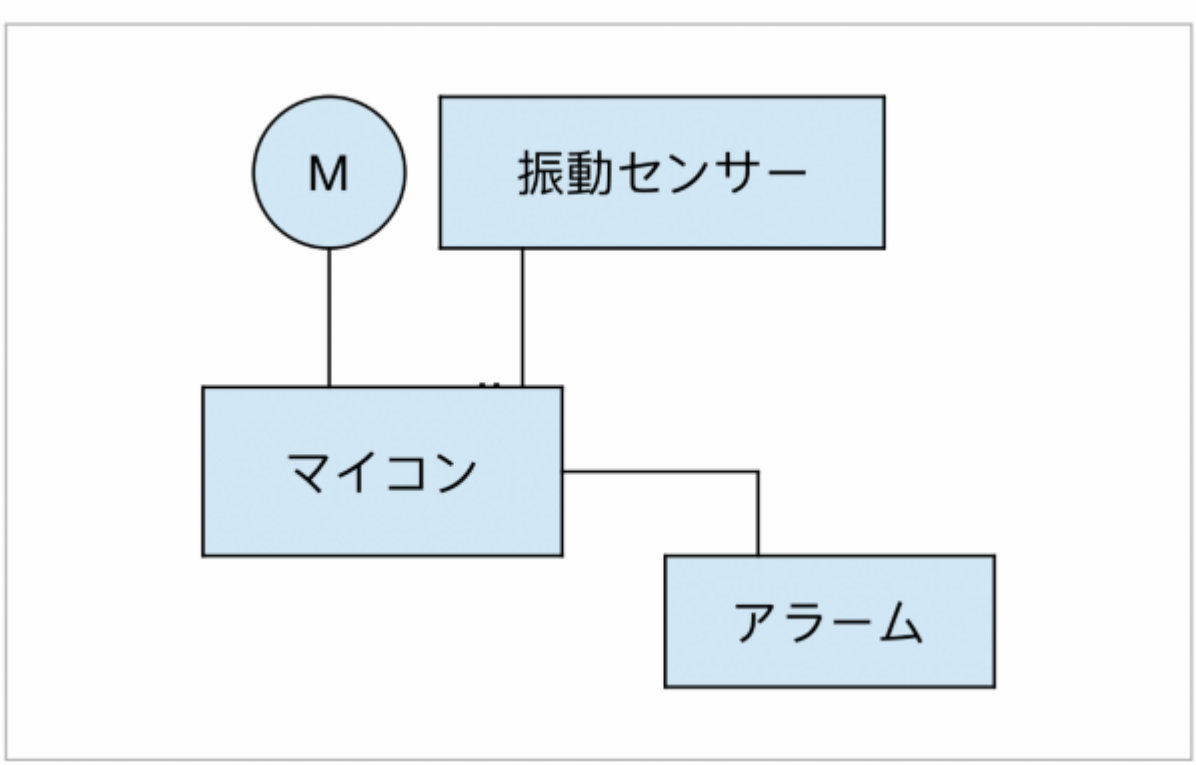
# 目的

## 1.モータ監視システム




### システムの動作:

1．モーターを駆動する。 2．振動センサーを動かす，モーターの状態を感知する。 3．モーターが意外で止まったら，アラームを鳴らす。 4.一秒以内でアラームを鳴らす。

## 2．主な仕様

| 仕様 | |
|---|---|
| マイコンのコア: | CotexM3 |
| DCモーター: | (3V) |
| アラーム: | 電子ブザー　１２mm　ＵＤＢ－０５ＬＦＰＮ |

# RTOSを作る

### 1．機能を実現ための関数を用意

```
//実際機能の関数を用意
void turnOnMotor(){  //モーターを駆動
        int output = 1;
}
int turnOnSensor(){  //センサーを駆動
        int output=1;
      return 1;//振動を感知してない場合は0を戻す
}
void turnOnAlarm(){
        int output =1;// アラームを駆動
}
```

ここでの関数は動作を示すための関数，実際は何もしない。

## 2．タスクを管理する関数

1．task1とtask2を用意

```
//タスク内容を保存するためのスタックを作る
tTask tTask1;
tTask tTask2;
tTaskStack task1Env[1024];
tTaskStack task2Env[1024];

// デフォルト タスク内容を保存するためのスタックを作る
tTask tTaskIdle;
tTaskStack idleTaskEnv[1024];
```

2．タスク配置するための変数を作る

```
// 今のタスク
tTask * currentTask;

// 次のタスク
tTask * nextTask;

// デフォルト タスク(何もしない)
tTask * idleTask;

// タスクリストのアレー(行列)を作る
tTask * taskTable[2];
```

3．タスク１と２に内容

```
//task1,2の機能：//変数Flagを0と1に繰り返し
//全部のタスクはCPU占有する，時間セグメントがきれると，次のタスクに切り替える。
int task1Flag;
void task1Entry (void * param)
{
    tSetSysTickPeriod(10);
      turnOnMotor();
    for (;;)
    {
        task1Flag = 1;  //Flagを 1 にする
        tTaskDelay(1);  //時間セグメントを待つ
        task1Flag = 0;  // Flagを 0 にする
        tTaskDelay(1);  //時間セグメントを待つ
    }
}

int task2Flag;
void task2Entry (void * param)
{
        if (turnOnSensor()==0){turnOnAlarm();}
        else{turnOnSensor();}

    for (;;)
    {
        task2Flag = 1;
        tTaskDelay(1);
        task2Flag = 0;
        tTaskDelay(1);
    }
}
```

4．デフォルトタスクの内容

```
//デフォルトタスク
void idleTaskEntry (void * param) {
    for (;;)
    {
        // 何もしない
    }
}
```

5.Main関数

```c
int main(){
    // タスク１２を初期化，最初のアドレスを転送する，タスクに任意の値を与える，スタックを配置する。
    tTaskInit(&tTask1, task1Entry, (void *)0x11111111, &task1Env[1024]);
    tTaskInit(&tTask2, task2Entry, (void *)0x22222222, &task2Env[1024]);

    // タスクリストにタスクを入れる。
    taskTable[0] = &tTask1;
    taskTable[1] = &tTask2;

    // デフォルト任務を作る
    tTaskInit(&tTaskIdle, idleTaskEntry, (void *)0, &idleTaskEnv[1024]);
    idleTask = &tTaskIdle;

    //まずはタスク１を実行する,最初で次のタスクをタスク１にする。
    nextTask = taskTable[0];

    // タスクをはじめる関数
    tTaskRunFirst();
    return 0;
    }
```

6.最初のタスクを実行

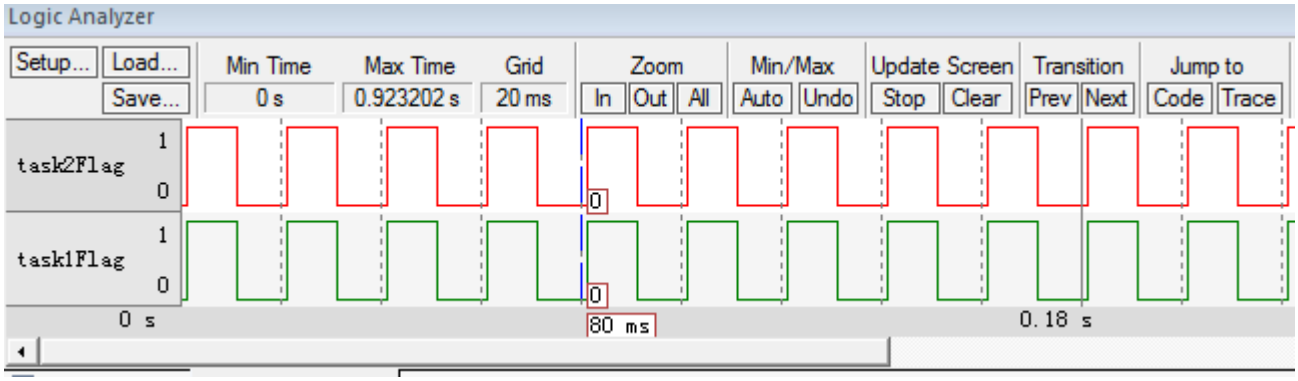
```c
//起動する時,一番目のタスク
void tTaskRunFirst()
{
    __set_PSP(0);
    //タスクの切り替えを行う関数（割り込みサービス）
    MEM8(NVIC_SYSPRI2) = NVIC_PENDSV_PRI;

    MEM32(NVIC_INT_CTRL) = NVIC_PENDSVSET;
}
```

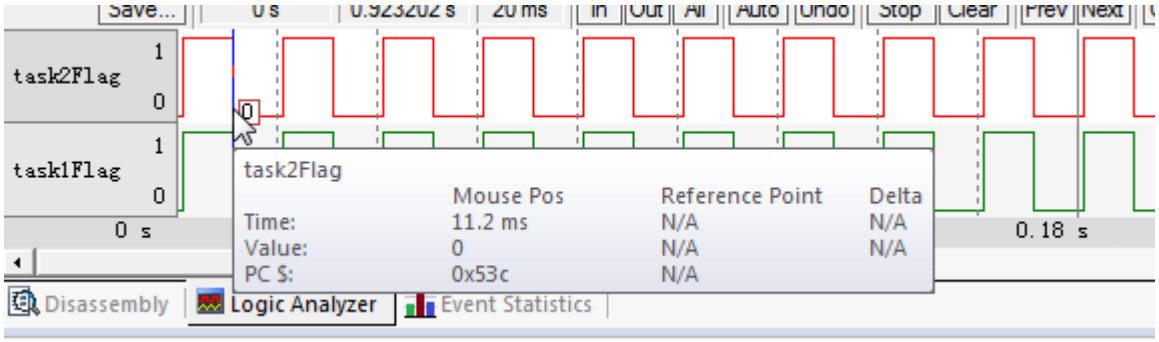
7.タスクを繰り返して切り替える

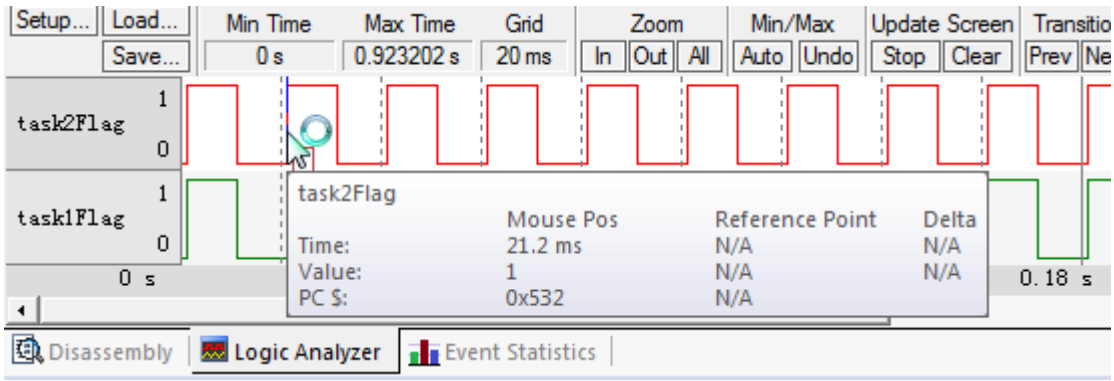
```c
//タスクを切り替え関数

void tTaskSwitch()
{

    MEM32(NVIC_INT_CTRL) = NVIC_PENDSVSET;
}
```

# 結果

10ms以内で二つのタスクが切り替えできます。ほぼ同時に運転してます。




# 参考文献

CotexM3の特徴　https://www.aps-web.jp/academy/cm/03/ http://01ketang.cc/usertos/task-delay-setting.html