

Bare Machine Design Example (4A)

Copyright (c) 2013 - 2016 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

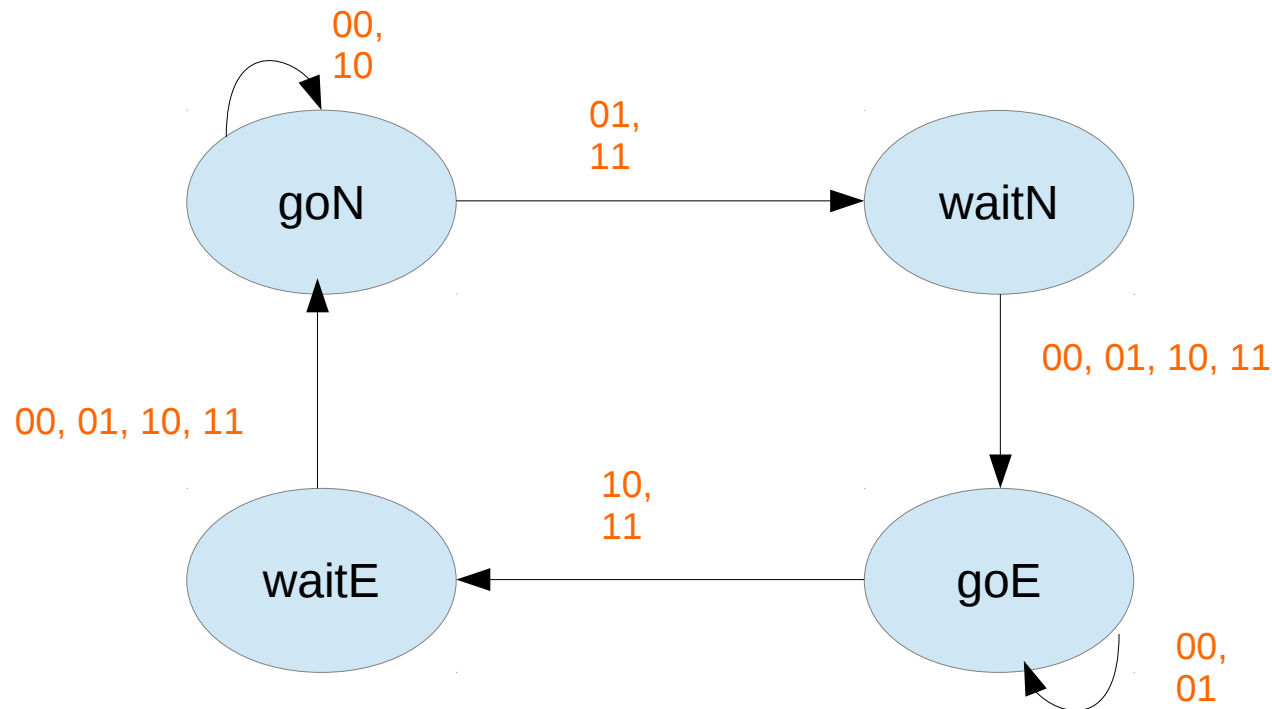
Main Infinite Loop

```
void main(void){  
  
    STyp *Pt; // state pointer  
  
    unsigned char Input;  
  
    Timer_Init();  
  
    DDRT = 0xFC; // lights and sensors  
  
    Pt = goN;  
  
    while(1){  
  
        PTT = Pt->Out << 2; // set lights  
  
        Timer_Wait10ms(Pt->Time);  
  
        Input = PTT & 0x03; // read sensors  
  
        Pt = Pt->Next[Input];  
  
    }  
}
```

http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C10_FiniteStateMachines.htm

FSM SW Implementation

		00,	01,	10,	11
goN	FSM[0] = {0x21, 300,	{goN,	waitN,	goN,	waitN}
waitN	FSM[0] = {0x22, 50,	{goE,	goE,	goE,	goE}
goE	FSM[0] = {0x04, 300,	{goE	goE,	waitE,	waitE}
waitE	FSM[0] = {0x14, 50,	{goN	goN,	goN,	goN}

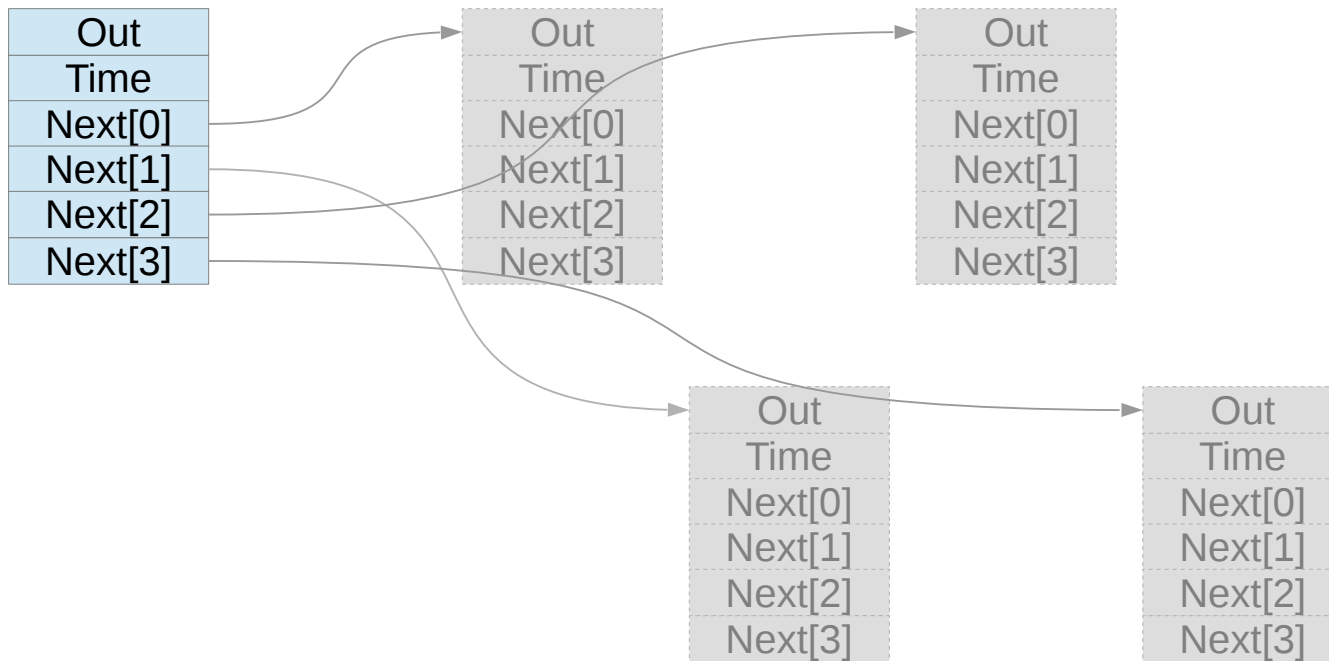


http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C10_FiniteStateMachines.htm

Struct State

```
struct State {  
    uint32_t Out;  
    uint32_t Time;  
    const struct State *Next[4];  
};
```

```
typedef const struct State STyp;
```

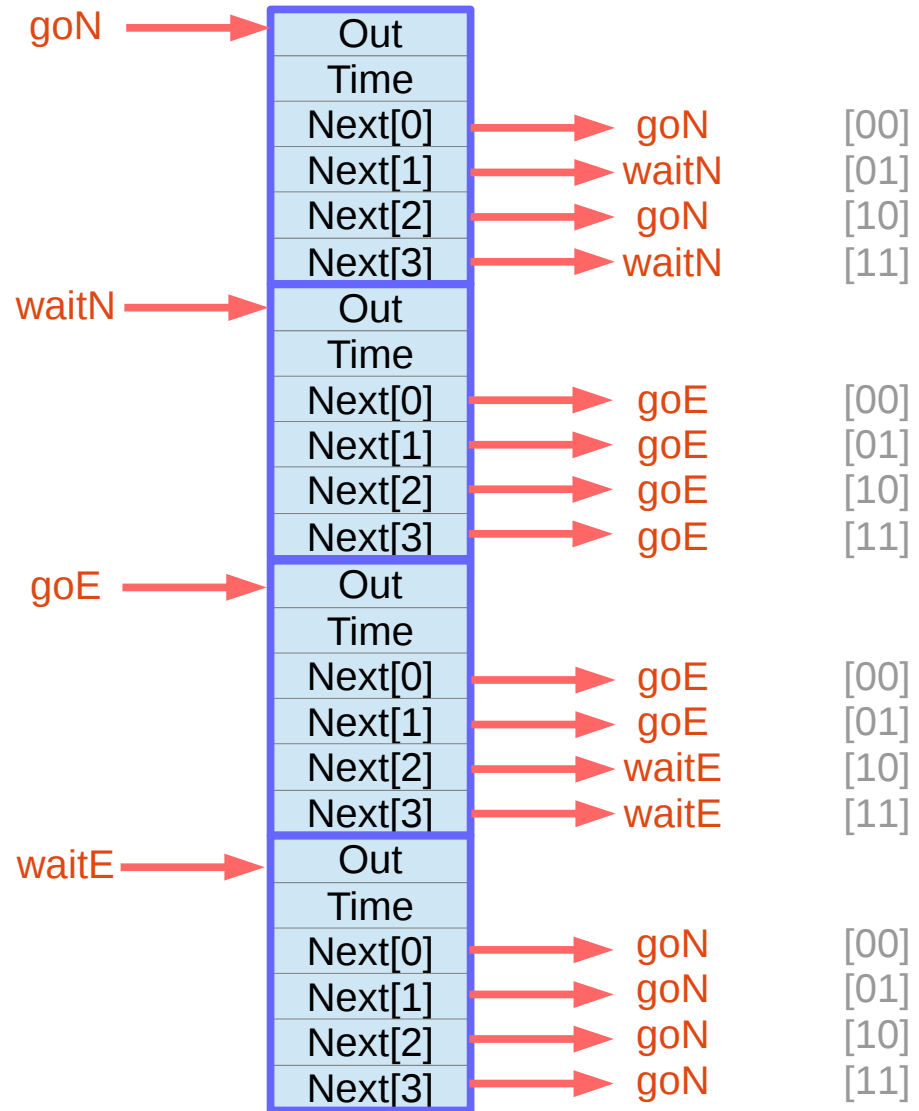


http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C10_FiniteStateMachines.htm

STyp FSM[4]

```
#define goN    &FSM[0]
#define waitN  &FSM[1]
#define goE    &FSM[2]
#define waitE  &FSM[3]
```

```
STyp FSM[4] = {
{0x21, 300, {goN,waitN,goN,waitN}},
{0x22, 50, {goE,goE,goE,goE}},
{0x0c, 300, {goE,goE,waitE,waitE}},
{0x14 50, {goN,goN,goN,goN}} }
```



http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C10_FiniteStateMachines.htm

Simple Index Implementation

```
struct State {
    unsigned long Out;
    unsigned long Time;
    unsigned long Next[4];};

typedef const struct State STyp;

#define goN    0
#define waitN  1
#define goE    2
#define waitE  3

STyp FSM[4]={
    {0x21,3000, {goN,waitN,goN,waitN}},
    {0x22, 500, {goE,goE,goE,goE}},
    {0x0C,3000, {goE,goE,waitE,waitE}},
    {0x14, 500, {goN,goN,goN,goN}}};

unsigned long S; // index to the current state
unsigned long Input;
```

```
unsigned long S; // index to the current state

int main(void){

    volatile unsigned long delay;

    S = goN;

    while(1){
        LIGHT = FSM[S].Out; // set lights

        SysTick_Wait10ms(FSM[S].Time);

        Input = SENSOR; // read sensors

        S = FSM[S].Next[Input];
    }
}
```

http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C10_FiniteStateMachines.htm

Assembly Implementation

```
org $0800
Pt      rmb 2      ;state pointer
;Linked data structure
org     $4000     ;Put in ROM
OUT     equ 0      ;offset for output
WAIT    equ 1      ;offset for time
NEXT    equ 3      ;offset for next
goN     fcb $21
        fdb 3000
        fdb goN,waitN,goN,waitN
waitN   fcb $22
        fdb 500
        fdb goE,goE,goE,goE
goE     fcb $0C
        fdb 3000
        fdb goE,goE,waitE,waitE
waitE   fcb $14
        fdb 500
        fdb goN,goN,goN,goN
```

```
main lds  #$4000    ;stack init
      bsr  Timer_Init ;enable TCNT
      ldaa #$FC     ;PT7-2 are lights
      staa DDRT    ;PT1-0 are sensors
      ldx  #goN    ;State pointer
      stx  Pt

FSM  ldx  Pt
      ldab OUT,x   ;Output value
      lslb
      lslb        ;line up with 7-2
      stab PTT     ;set lights
      ldy  WAIT,x  ;Time delay
      bsr  Timer_Wait10ms
      ldab PTT     ;Read input
      andb #$03   ;just bits 1,0
      lslb        ;2 bytes/address
      abx        ;add 0,2,4,6
      ldx  NEXT,x  ;Next state
      stx  Pt
      bra  FSM
      org  $FFFE
      fdb  main    ;reset vector
```

http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C10_FiniteStateMachines.htm

Simple Index Implementation

```
struct State {
    unsigned long Out;
    unsigned long Time;
    unsigned long Next[4];};

typedef const struct State STyp;

#define goN    0
#define waitN  1
#define goE    2
#define waitE  3

STyp FSM[4]={
    {0x21,3000, {goN,waitN,goN,waitN}},
    {0x22, 500, {goE,goE,goE,goE}},
    {0x0C,3000, {goE,goE,waitE,waitE}},
    {0x14, 500, {goN,goN,goN,goN}}};

unsigned long S; // index to the current state
unsigned long Input;
```

```
unsigned long S; // index to the current state

int main(void){

    volatile unsigned long delay;

    S = goN;

    while(1){
        LIGHT = FSM[S].Out; // set lights

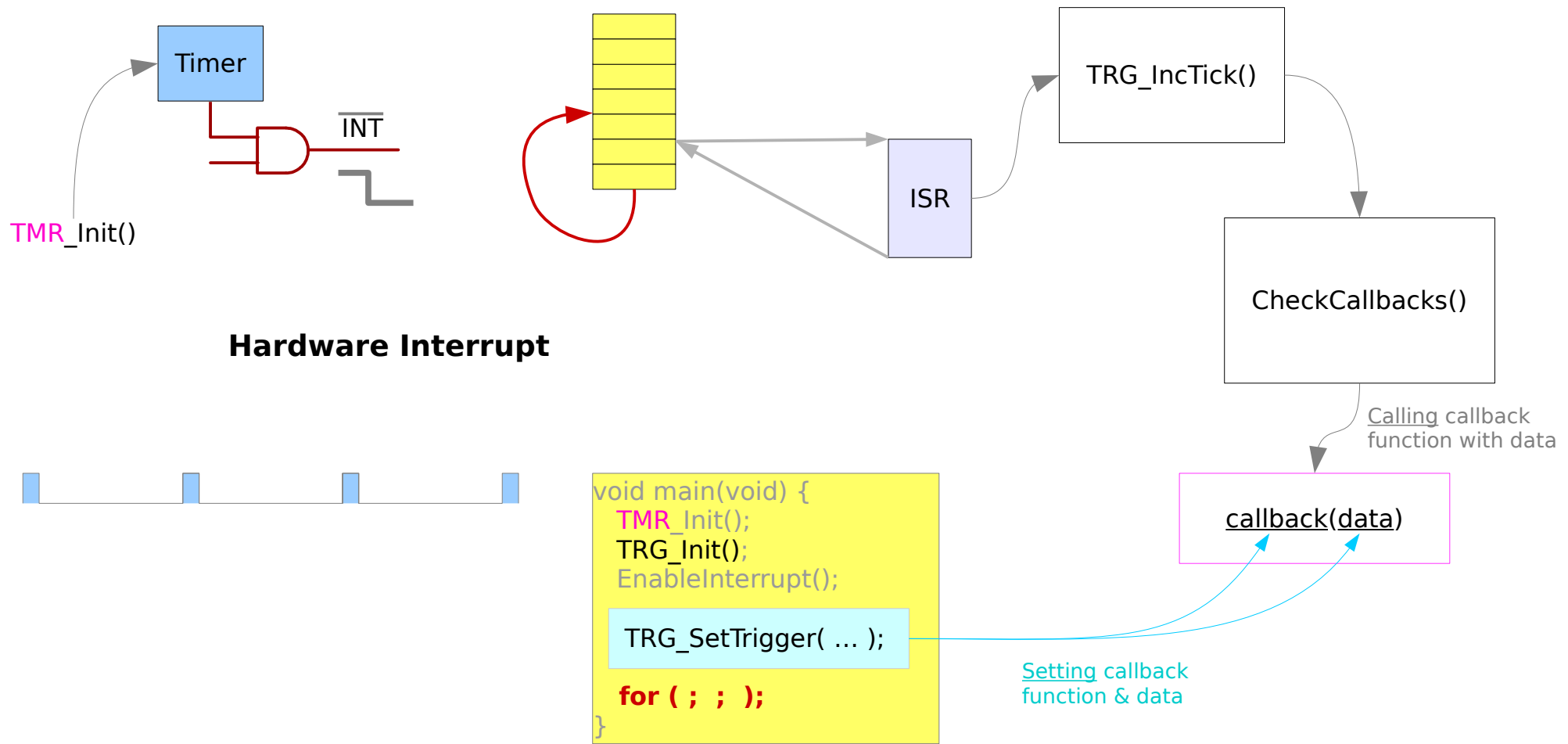
        SysTick_Wait10ms(FSM[S].Time);

        Input = SENSOR; // read sensors

        S = FSM[S].Next[Input];
    }
}
```

http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C10_FiniteStateMachines.htm

Timer & Interrupt



Critical Section Access

```
TRG_Trigger[i].ticks = ticks;  
TRG_Trigger[i].callback = callback;  
TRG_Trigger[i].data = data;
```

```
(TRG_Trigger[i].ticks != 0)
```

```
TRG_Trigger[i].ticks--;
```

```
TRG_SetTrigger( ... );
```

WR

RD/WR

```
TRG_IncTick()
```

```
CheckCallbacks()
```

Shared Data

```
TRG_Trigger
```

	tick	callback	data
0			
1			
2			
3			

RD/WR

```
(TRG_Trigger[i].ticks == 0)  
(TRG_Trigger[i].callback != NULL)
```

```
callback = TRG_Trigger[i].callback;  
data = TRG_Trigger[i].data;  
TRG_Trigger[i].callback = NULL;
```

```
static TRG_TriggerDesc TRG_Triggers[TRG_NOF_TRIGGERS];
```

File Scope : static global variable

References

- [1] <http://en.wikipedia.org/>
- [2] D.M. Harris, S. L. Harris, "Digital Design and Computer Architecture"