

Lab 5 Lecture

Monday, March 1, 2021 6:22 PM

Lab 5 is all in C!

Topics:

- ① SysTick
- ② Structs inc
- ③ PSM
- ④ FSM's in C

Systick:

Systick is a counter exist on the TM4C

Properties:
- 24 bits counter
- Counts down at a bus clock
- 80 MHz \rightarrow 12.5 ns per decrement

Address	31- 24	23- 17	16	15- 3	2	1	0	Name
\$E000E010	0	0	COUNT	0	CLK_SRC	INTEN	ENABLE	NVIC_ST_CTRL_R
\$E000E014	0			24-bit RELOAD value				NVIC_ST_RELOAD_R
\$E000E018	0			24-bit CURRENT value of SysTick counter				NVIC_ST_CURRENT_R

```
void SysTick_Init(void){  
    NVIC_ST_CTRL_R = 0; // 1) disable SysTick during setup  
    NVIC_ST_RELOAD_R = 0x00FFFFFF; // 2) maximum reload value  
    NVIC_ST_CURRENT_R = 0; // 3) any write to current clears it  
    NVIC_ST_CTRL_R = 0x00000005; // 4) enable SysTick with core clock  
}
```

Structs: A way to organize data w/ different types together

Struct player {

```
    int8_t points;  
  
    char* name;  
  
    uint32_t goals[3];  
};  
  
type def struct player player_t;
```

Player-t playerOne = {0, "John", {10, 20, 30}};

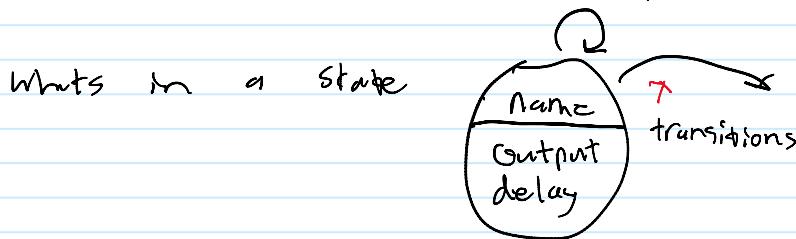
Player-t playerArray[3] = {
 {1, "Sam", {100, 50, 23},
 {2, "Bob", {7, 18, 123},
 {27, "Jim", {100, 50, 28}}
 }
}

To access elements of structs

int8-t topscore = playerOne.points;
" " = playerArray[2].points;
= 27

FSM: Moore FSM's

Key Aspects: output (current state)
transitions (inputs and current state)



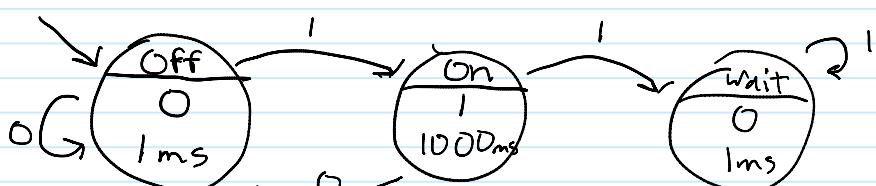
Struct State {

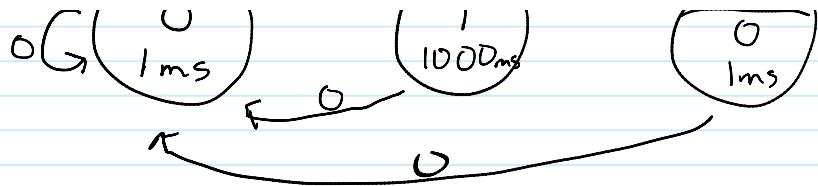
int32-t out;
int32-t delay; where x is input bits
int32-t next[2^x];

3 types of struct state = state-t;

① Capture the behavior

Problem: push of a button activates LED for
ONLY 1 sec (1000ms)





(2) Define States

Struct State {

```
int32_t out;
int32_t delay;
int32_t next[2];
```

```
3
typedef struct state state_t;
```

(3) #define off 0

#define on 1

#define wait 2

state_t PSIN[3] = {

{ 0, 1, {off, on} },

{ 1, 1000, {off, wait} },

{ 0, 1, {off, wait} },

}

(4) Create the engine

Pattern : ① Perform output of c.s.

② wait a specified amount of time

③ Input

④ Transition to next state

How do we do this in code?

Unt32_t cs = off;

```

        V int8_t input;
    while(1) {
        // Perform output of CS.
        // wait a specified amount of time
        // get input
        3 // transition states
    }

```

Input: PE0, Output: PA1

while(1) {

 GPIO_PortB_Data_h = FSM[cs].out; } ^{FSM}_{dependent}

 SysTick_lms(FSM[cs].delay);

 input = (GPIO_PortA_Data_h && 0x02) >> 1;

 cs = FSM[cs].next[input]; }

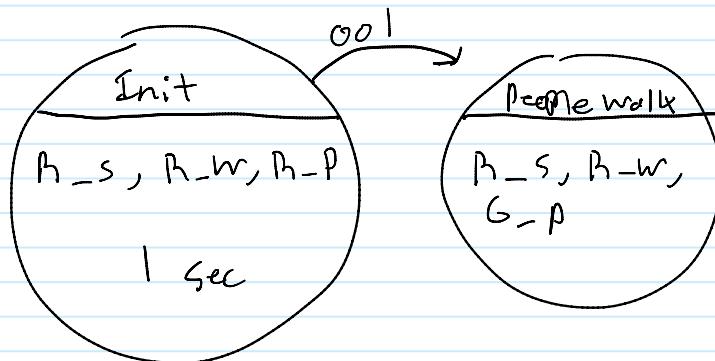
For Lab 5: Inputs: cars-S, cars-W, people → binary #

Outputs: R-S, Y-S, G-S

R-W, Y-W, G-W

R-P, Y-P, G-P

} non-binary



Circuit for the lab

PF1 → Don't walk

PF3 → Walk

PF3 → Walk

