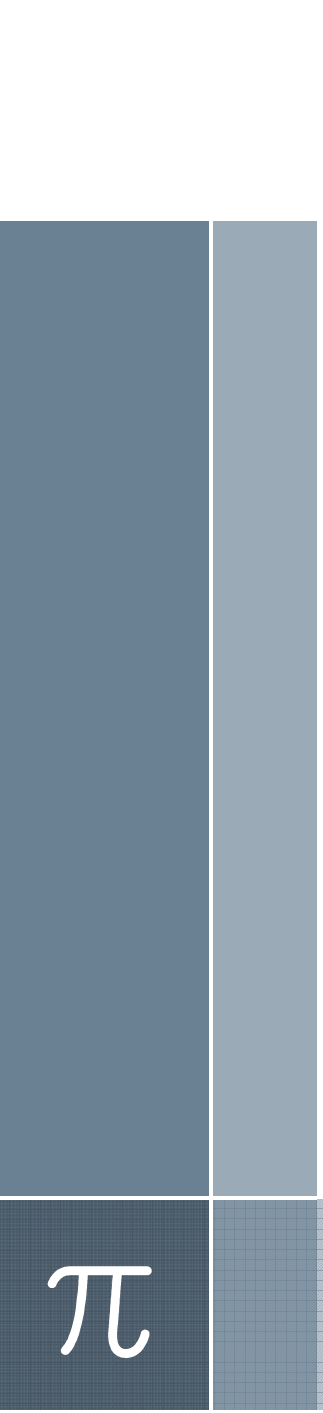
Embedded Thursday

Binary, Registers and Blinky



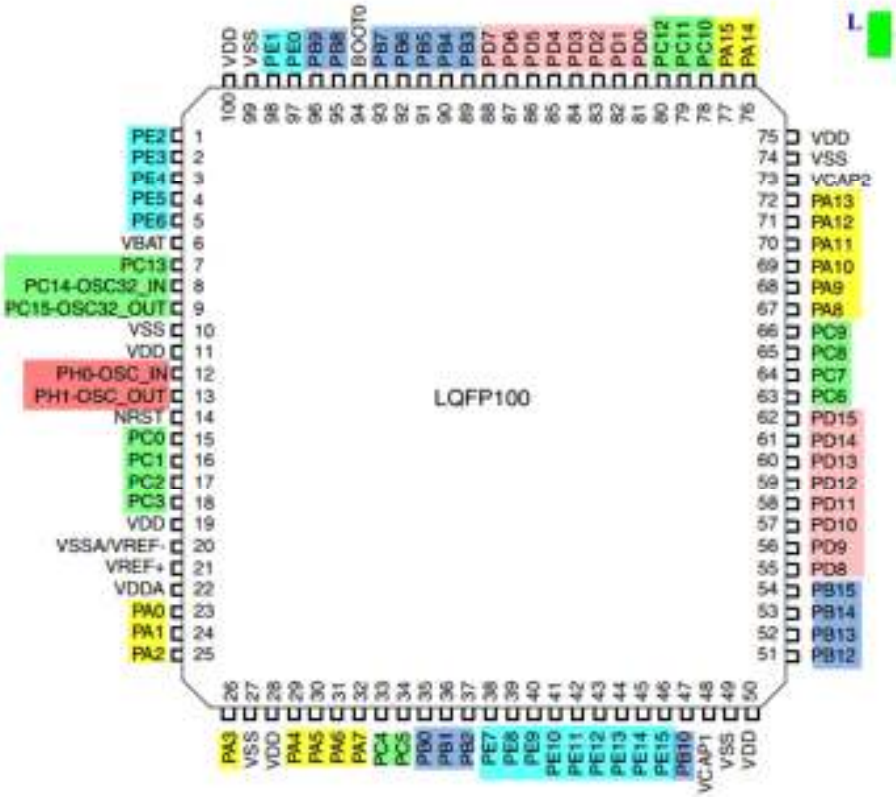
Today

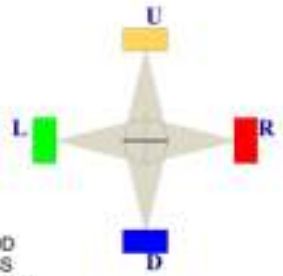
- The project Goal and System
- Registers Discussion
- Binaries
- Blinky

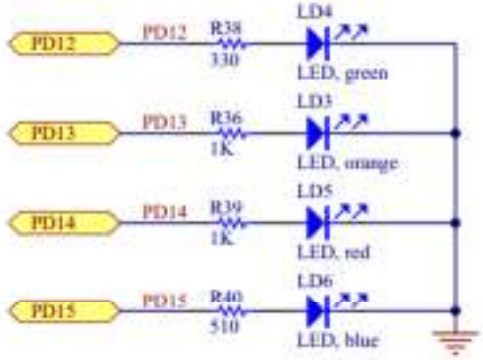
Goal Description

- Learn C as embedded language
- Use C to understand underlying processor
- Have a project so learning stays
  - We are making a self balancing robot

STM32F411E







- Port-A: 0-15
- Port-B: 0-10, 12-15
- Port-C: 0-15
- Port-D: 0-15
- Port-H: 0-1

Why Registers?

- It allows us to set up the configuration for the controller
- It allows us to control ports

6.3.9 RCC AHB1 peripheral clock enable register (RCC_AHB1ENR)

Address offset: 0x30

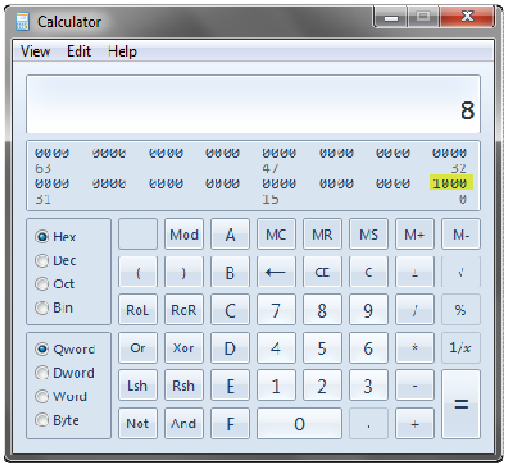
Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									DMA2EN	DMA1EN	Reserved				
									rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CRCEN	Reserved				GPIOH EN	Reserved		GPIOEEN	GPIOD EN	GPIOC EN	GPIOB EN	GPIOA EN
			rw					rw			rw	rw	rw	rw	rw

Bit 3 **GPIODEN:** IO port D clock enable
Set and cleared by software.
0: IO port D clock disabled
1: IO port D clock enabled



- RCC_AHB1ENR = 0x00000008 Enable GPIO D
- RCC_AHB1ENR = 0x00000001 Enable GPIO A
- RCC_AHB1ENR |= 0x00000008 Add GPIO A enable

Registers Expected by microcontroller

- At power up all GPIOs are disabled
  - To blink an LED enable PORT where LED are connected
- GPIO PORT enabled can be INPUT or OUTPUT
  - Set LEDs to be OUTPUTS
  - IN/OUT Register control is GPIOx_MODER

8.4.1 GPIO port mode register (GPIOx_MODER) (x = A..E and H)

Address offset: 0x00

LED6 LED5 LED3 LED4

D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
0	1	0	1	0	1	0	1								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 2y:2y+1 **MODERy[1:0]:** Port x configuration bits (y = 0..15)

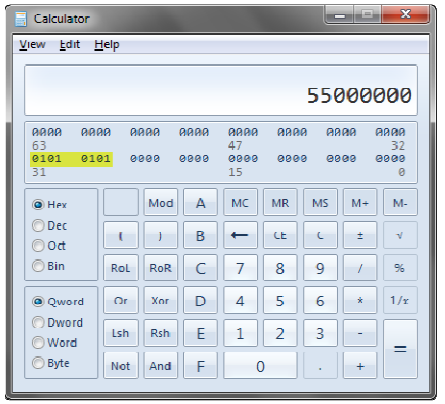
These bits are written by software to configure the I/O direction mode.

00: Input (reset state)

01: General purpose output mode

10: Alternate function mode

11: Analog mode



- GPIOD_MODER = 0X55000000

Operators Used

- XOR ^= (switch bit to opposite value)
- OR |= (Impact a bit, don't disturb others)
- Member of -> (similar to house.room)

- 0100 ^= 0001 ➔ Results in 0101
- 1101 |= 0100 ➔ Results in 1001
- 1001 |= 0100 ➔ Results in 1101
- RCC->AHB1ENR ➔ refers to RCC_AHB1ENR where RCC is a pointer in memory for the register

Blinky

```
12  #include "stm32f4xx.h"
13  #include "stm32f411e_discovery.h"
14
15  int main(void) {
16  //  RCC->AHB1ENR |= RCC_AHB1ENR_GPIODEN; // enable the clock to PORT-D
17      RCC->AHB1ENR |= 0x00000008; // enable the clock to PORT-D
18      GPIOD->MODER = 0X55000000;   // Set Port-D pin12 to 14 to OUTPUTS
19
20  /*  GPIOD->MODER |= (1 << 24); // set pin 12 to be general purpose output..
25      volatile int i;
26
27      while (1){
28  /*      GPIOD->ODR ^= (1 << 12);..
33          GPIOD->ODR ^= 0x0000F000;      //
34          for (i = 0; i < 500000; i++);
35          }
36  }
```