EECS192 Lecture 12
Apr. 11, 2017

Notes:

1. Mon. 4/25: **(530-700 pm**) round 2 (NATCAR rules)
    1. 13 makes first turn
    2. 15 half track in < 5 minutes
    3. 18 whole track in less than 1 minute
    4. > 18 For cars which are fast and/or well-stabilized
2. CalDay Sat. April 22 @ UCB, 1000-1200 SDH (in Cory class room if rain)
3. Quiz 5 on 4/11 on steering control

Topics

- Round 1 results
- Progress report notes: Hamamatsu
- Hardware Robustness
- C.O.P. Watchdog
- Software Robustness
- Supervisor Systems
- Digital Filtering

Spring 2017 Round 1 Results

¾ track (8.5 pts): team 1, 10
½ track (8 pts): team 8,9
¼ track (7.5): team 5,12
2 turns, or turn+steps (7 pts):2,3
1 turn (6.5 pts): 4,6,7
Moves (6 pts): 11

Lessons Learned

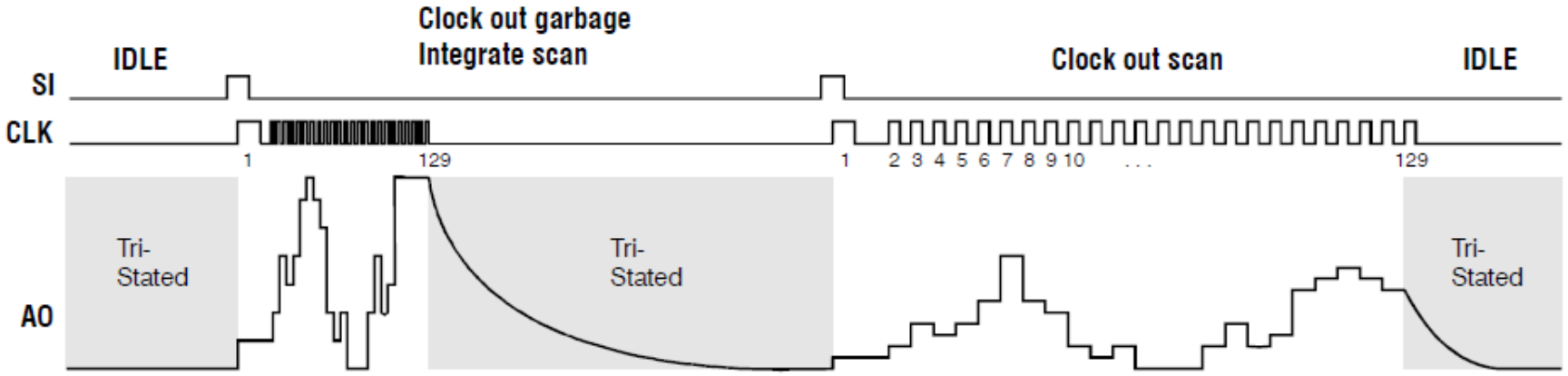
Lessons Learned:

- Extreme lighting variation ➔
  Need AGC,
  steering hold,
  Autocalibration
  Sun glasses …
- Camera mechanics
- ? Camera Noise level?
- Track vs terrain features
- Self-shadow

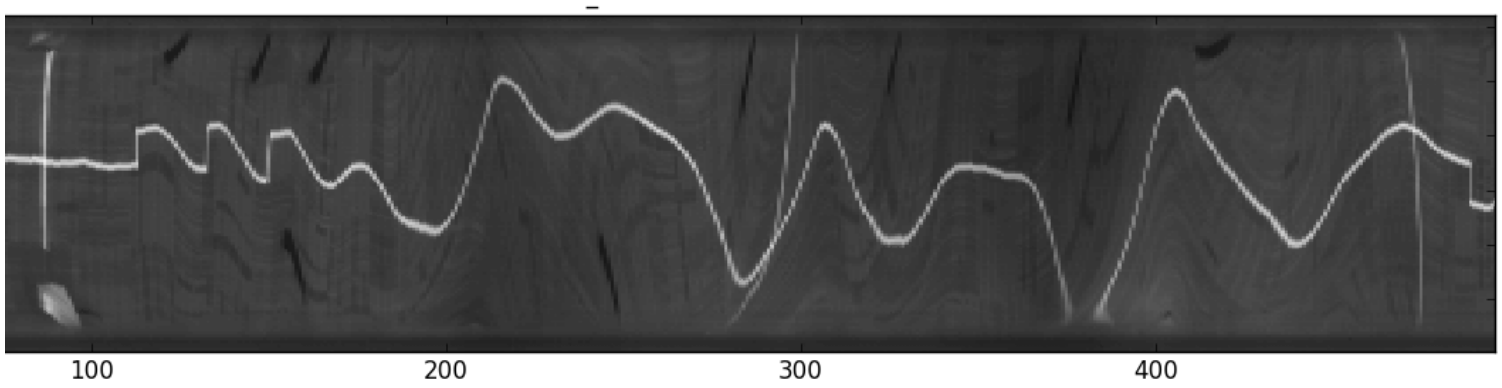
What worked well:

- Velocity control
- Stability decent
- Car electromechanics
- No DoA cars
- No random resets or runaway cars
- No tank cars
- Crossings ok vs crack not

Automatic Gain Control



In all the discussion that follows, we will be using one-shot imaging.



Progress report notes

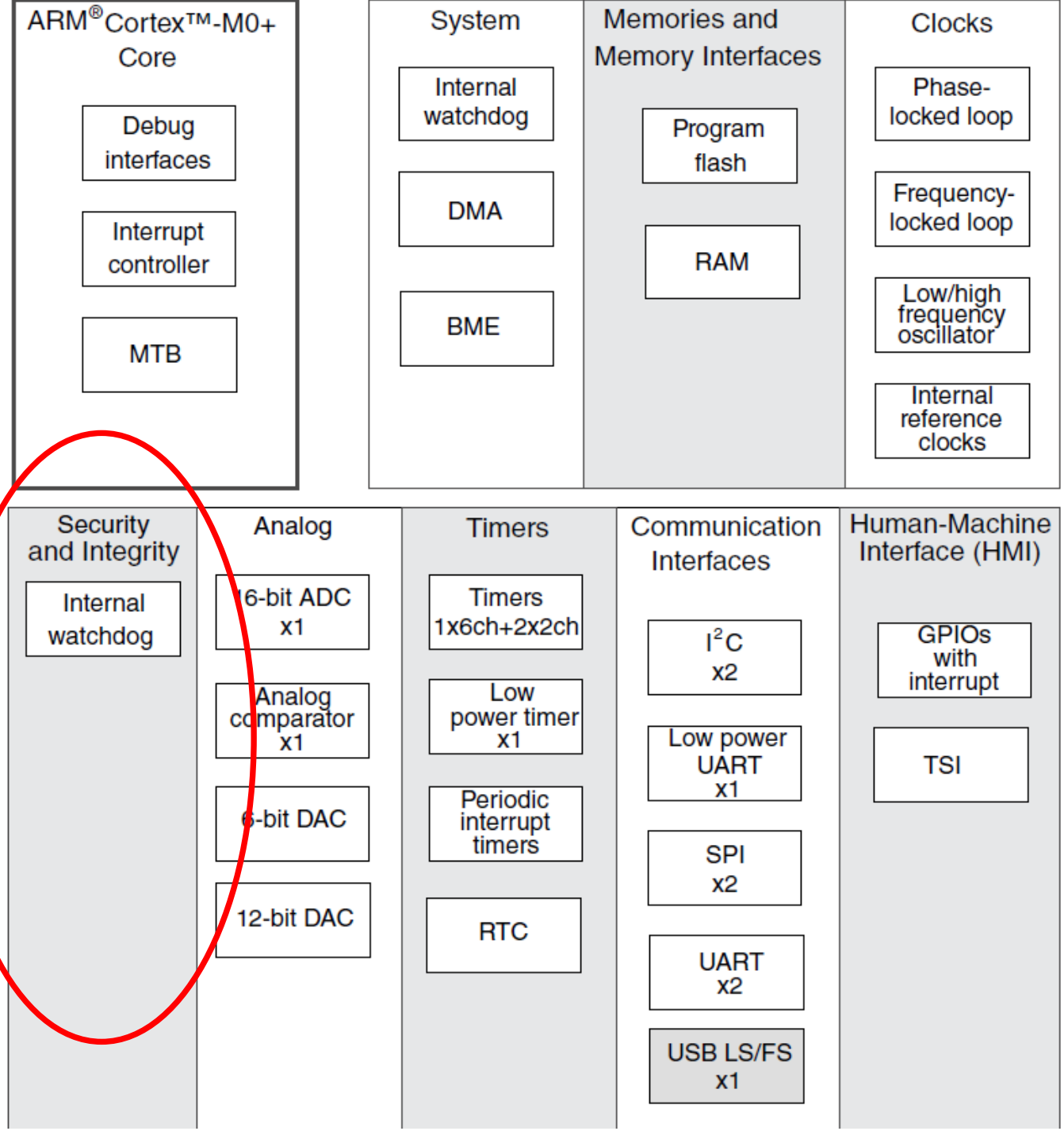
- Hamamatsu: 5 volts, be careful
- Motor snubber caps

Hardware Robustness

- Mechanical oscillations
- Lock washers
- Strain relief on connections
  - (stranded vs solid core wire)

C.O.P. Watchdog timer

- Despite extensive software and hardware testing, faults will still occur in real devices. Even momentary noise spikes on a power supply can lock up a processor occasionally. Such events will occur on the power grid several times a year. Watchdog timers provide a last line of defense to prevent system failure with minimal hardware cost.
- https://developer.mbed.org/cookbook/Watch Dog-Timer



128K Flash
16K RAM
32 bit ARM 7 core
48 MHz
A/D, D/A
2x SPI
Touch sense input
Timers

3.4.10 Computer Operating Properly (COP) Watchdog Configuration

This section summarizes how the module has been configured in the chip.

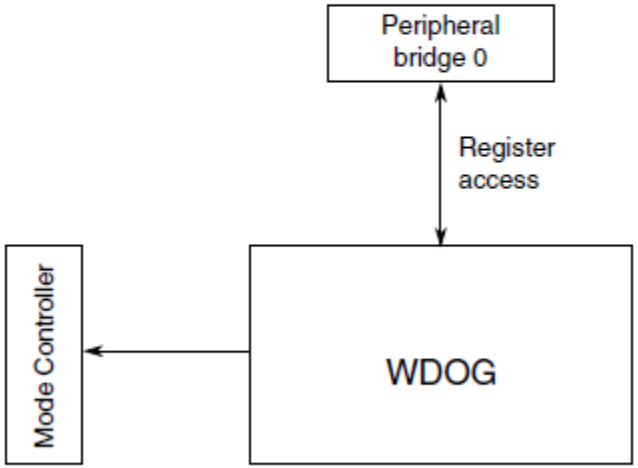


Figure 3-13. COP watchdog configuration

Address: 4007_F000h base + 0h offset = 4007_F000h

Bit	7	6	5	4	3	2	1	0
Read	POR	PIN	WDOG	0	LOL	LOC	LVD	WAKEUP
Write								
Reset	1	0	0	0	0	0	1	0

RCM_SRS0 field descriptions

Field	Description
7 POR	Power-On Reset Indicates a reset has been caused by the power-on detection logic. Because the internal supply voltage was ramping up at the time, the low-voltage reset (LVD) status bit is also set to indicate that the reset occurred while the internal supply was below the LVD threshold. 0 Reset not caused by POR 1 Reset caused by POR
6 PIN	External Reset Pin Indicates a reset has been caused by an active-low level on the external $\overline{\text{RESET}}$ pin. 0 Reset not caused by external reset pin 1 Reset caused by external reset pin
5 WDOG	Watchdog Indicates a reset has been caused by the watchdog timer Computer Operating Properly (COP) timing out. This reset source can be blocked by disabling the COP watchdog: write 00 to the SIM's COPC[COPT] field. 0 Reset not caused by watchdog timeout 1 Reset caused by watchdog timeout
4	This field is reserved

Table 3-23. COP configuration options (continued)

Control Bits		Clock Source	COP Window Opens	COP Overflow Count
COPCTRL[COPCLKS]	COPCTRL[COPT]		(COPCTRL[COPW]=1)	
0	10	1 kHz	N/A	2^8 cycles (256 ms)
0	11	1 kHz	N/A	2^{10} cycles (1024 ms)
1	01	Bus	6,144 cycles	2^{13} cycles
1	10	Bus	49,152 cycles	2^{16} cycles
1	11	Bus	196,608 cycles	2^{18} cycles

Need to change systemInit:

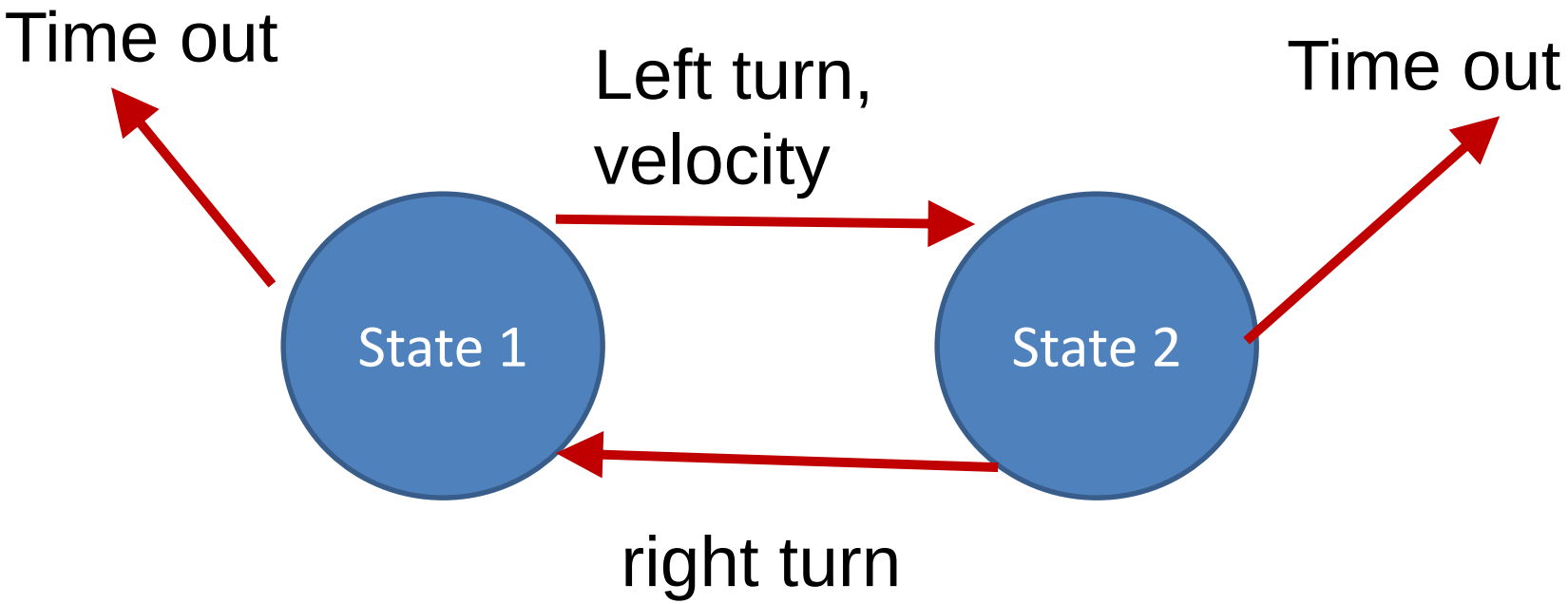
```
void SystemInit (void)
{ #if (DISABLE_WDOG)   /* Disable the WDOG module */
 /* SIM_COPC: COPT=0,COPCLKS=0,COPW=0 */
SIM->COPC = (uint32_t)0x00u;
#endif /* (DISABLE_WDOG) */
```

```
// Kick (feed, reload) our watchdog timer
 void wdt_kick()
{
 SIM->SRVCOP = (uint32_t)0x55u;
 SIM->SRVCOP = (uint32_t)0xAAu;
 }
```

Software Robustness

- Checksums for bit rot
- Watch dog timer/computer operating properly COP
- Lost track detection
- Autocalibration at startup
  - (sanity check for steering angle vs line error)
  - AGC
- State Observer/estimator
- Discrete State observer

FSM Recognizer (generalized WDT)



Digital Filtering

- Moving average
  - $y1[n] = (y[n-2]+y[n-1]+y[n])/3$
- Median filter (outlier rejection)
- Notch filter (mechanical vibration)
  - $y[n] = (x[n-2]+2x[n-1]+x[n])/4$
- Model based filtering (or Kalman filter)

(on board)

Round 1 Results

- **2015**: 48.14, 46.56, 67.93, DNF, 53.72, 85.51, 40.37, 40.24, 41.61, DNF, 62.88, 45.59
- **2016**: 33.61, 42.32, 42.48, 42.83, 55.14, 62.59, 66.72, 67.69, 140.01, DNF,