EECS192 Lecture 4
Velocity and Line Sensor I
Feb. 11, 2020

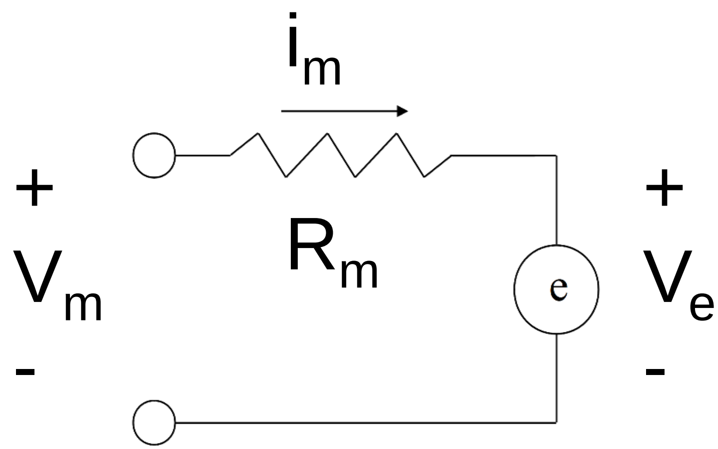
Notes:
Check off-
- 2/14: Motor drive/stall, steering servo using LiPo
- Quiz 2: power MOSFET/motor drive Tues 2/18

Topics
- Driving MOSFETs and motor (conclusion)
- Battery Model
- Power wiring
- Driving MOSFETs and motor (conclusion)
- Quiz 1
- Latchup gotcha
- Speed sensing
- Line sensor

Topics

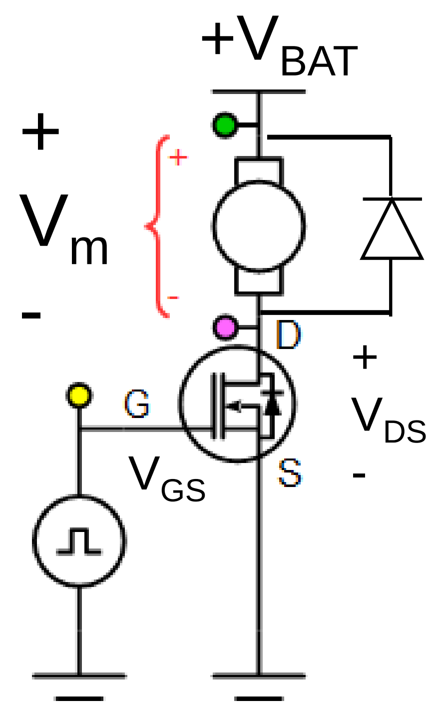
- Driving MOSFETs and motor (conclusion)
- Battery Model
- Power wiring
- Quiz 1
- Latchup gotcha
- Speed sensing
- Line sensor

Motor Electrical Model



Motor Electrical Model
Back EMF
Motor electromechanical behavior

Also- see motor worksheet……



$$i_m = \frac{V_{BAT} - k_e \dot{\theta}_m}{R_m}$$

Motor Resistance?
Peak current?

Conclusion:
$\langle i_m \rangle = ?$

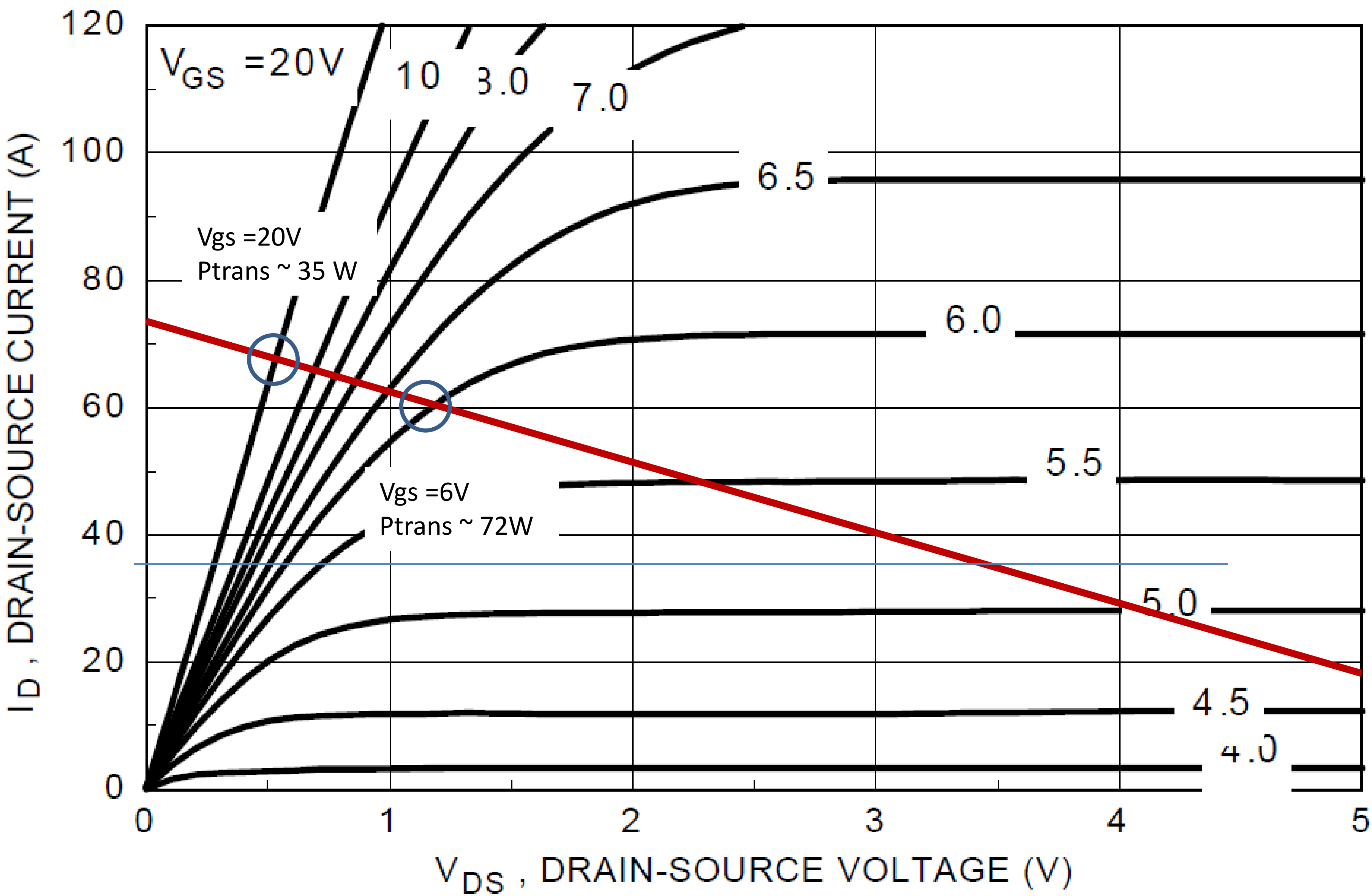
Driving MOSFETs and motor

Rm = 0.1 ohms, Vbatt = 7.2 V, Rbat = 0.

Vds = 3.6V ➔ Ids = (7.2-3.6V)/(0.1 ohm) = 36 amps

Key design points:

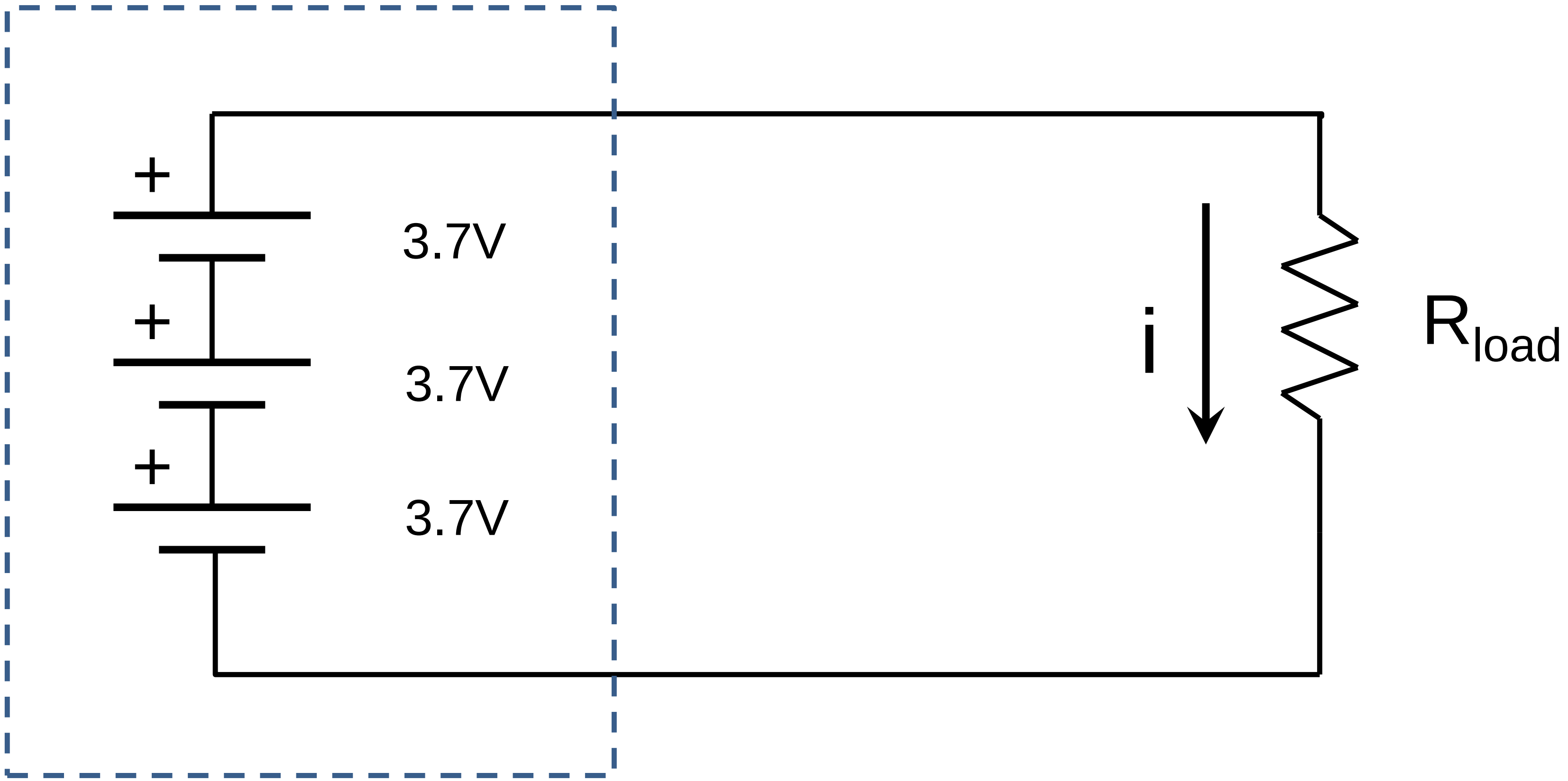
1) High Vgs better than low Vgs
2) Switch quickly
3) Make sure Vs=0 (big ground)



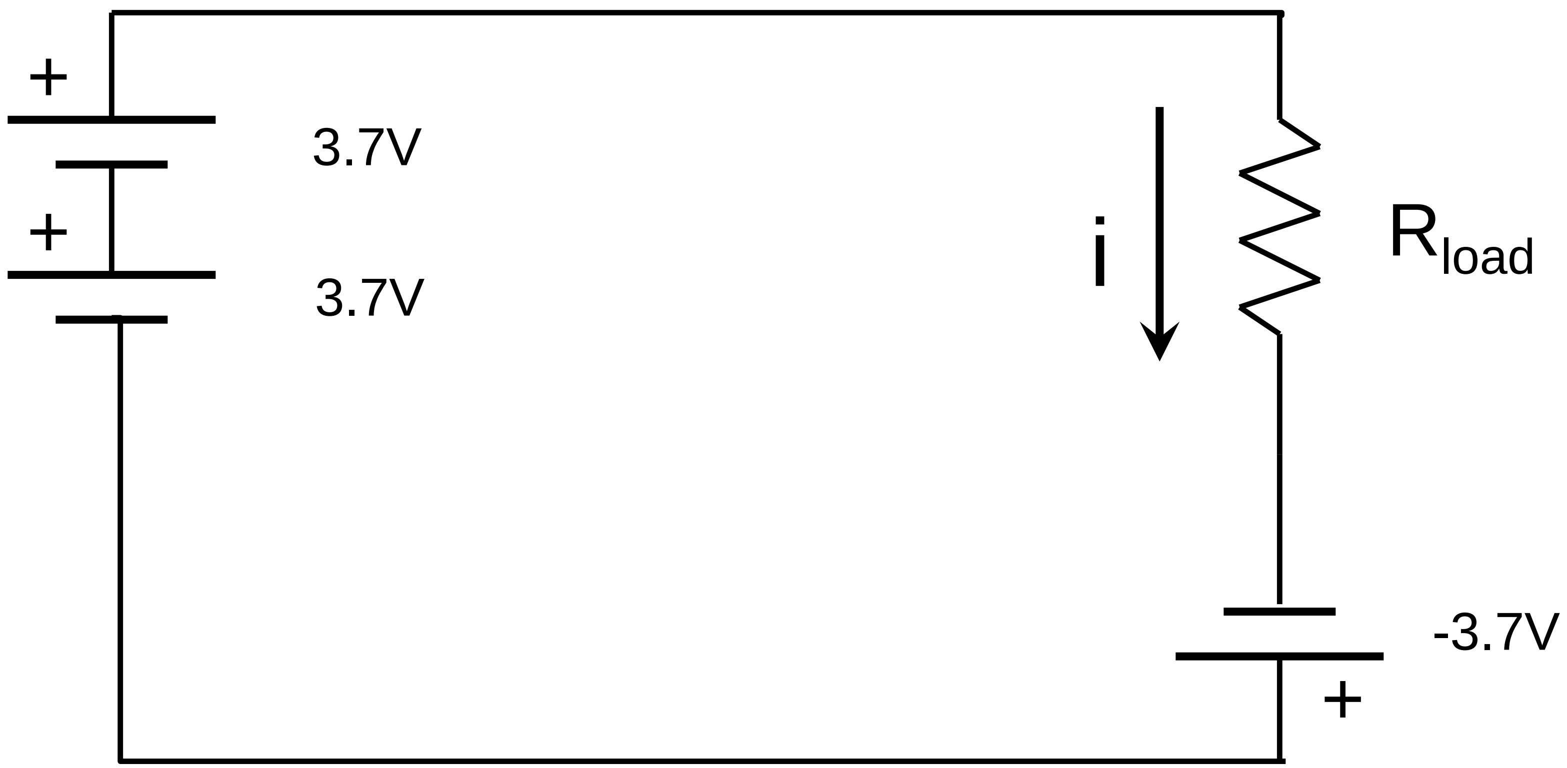
Topics

- Driving MOSFETs and motor (conclusion)
- Battery Model
- Power wiring
- Quiz 1
- Latchup gotcha
- Speed sensing
- Line sensor

Battery Model- 3S



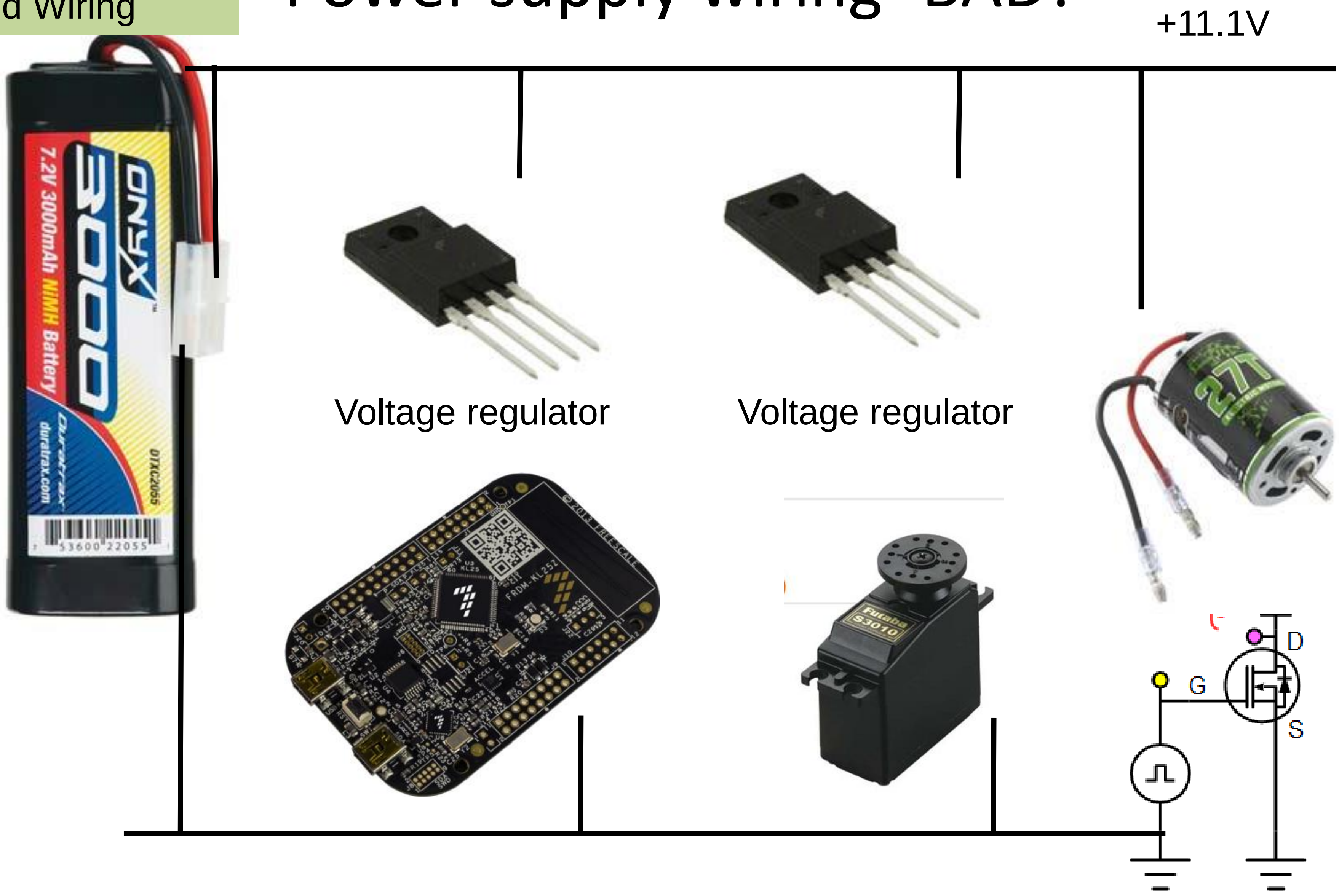
Battery Model- 3S
avoid weakly charged cell



Topics

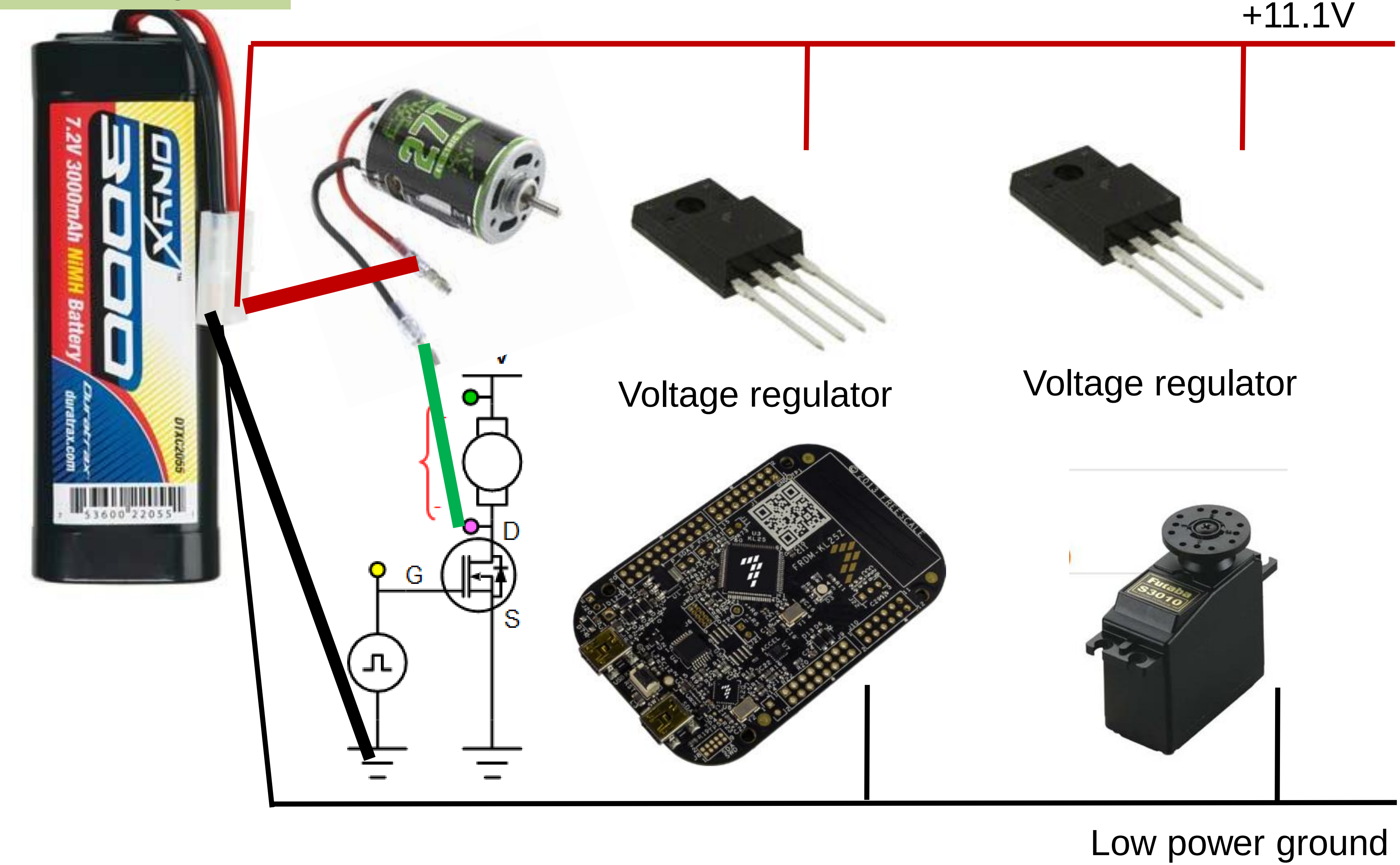
- Driving MOSFETs and motor (conclusion)
- Battery Model
- Power wiring
- Quiz 1
- Latchup gotcha
- Speed sensing
- Line sensor

Power supply wiring- BAD!



On board: what does this look like electrically (as a schematic)?

Power supply wiring- Star is better!



On board: what does this look like electrically (as a schematic)?

Topics

- Driving MOSFETs and motor (conclusion)
- Battery Model
- Power wiring
- Quiz 1
- Latchup gotcha
- Speed sensing
- Line sensor

1.4 Voltage and current operating ratings

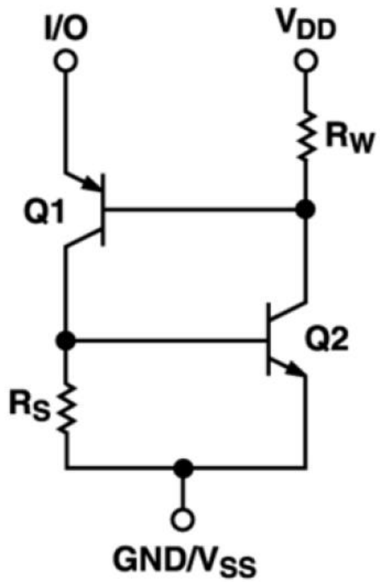
Table 4. Voltage and current operating ratings

LATCHUP!

Symbol	Description	Min.	Max.	Unit
V_{DD}	Digital supply voltage	–0.3	3.8	V
I_{DD}	Digital supply current		1[illegible]0	mA
V_{IO}	IO pin input voltage	–0.3	V_{DD} + 0.3	V

Caution: input voltage from sensor may be greater than 0.3V when CPU is off
VDD = 0!

Latchup phenomena:
make sure Vin always less than Vdd



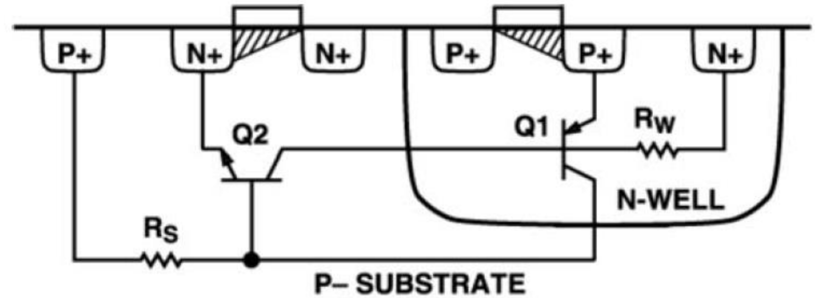
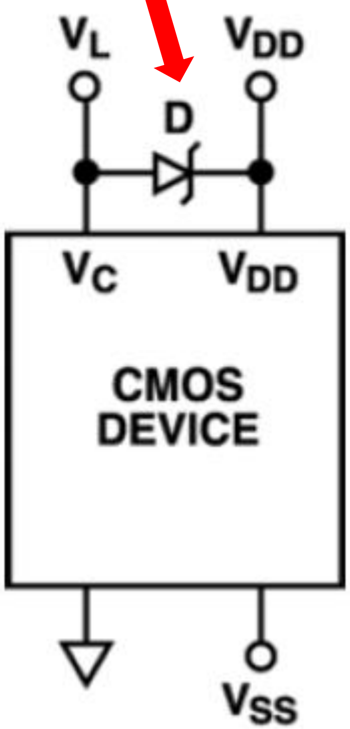


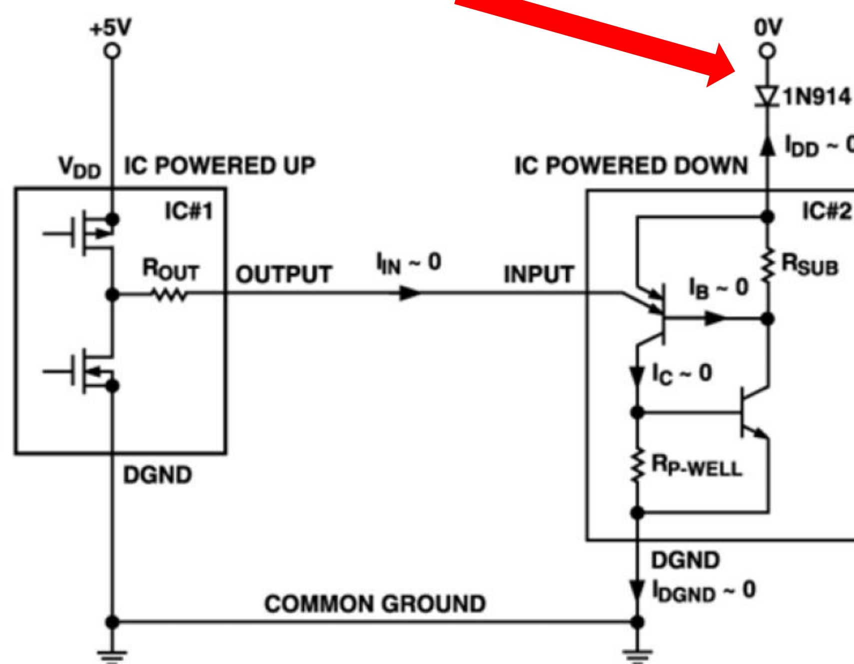
Figure 2. Cross-section of PMOS and NMOS devices, showing parasitic transistors Q1 and Q2.

Protection circuit



Schottky diode

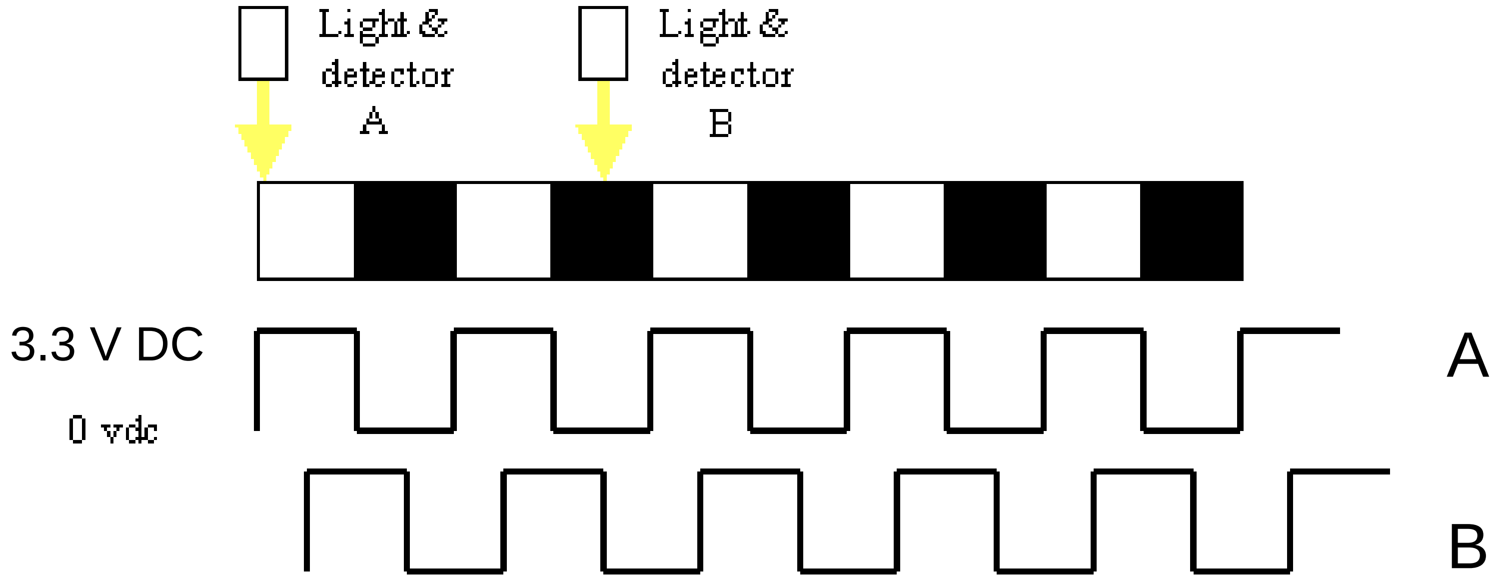
Protection circuit



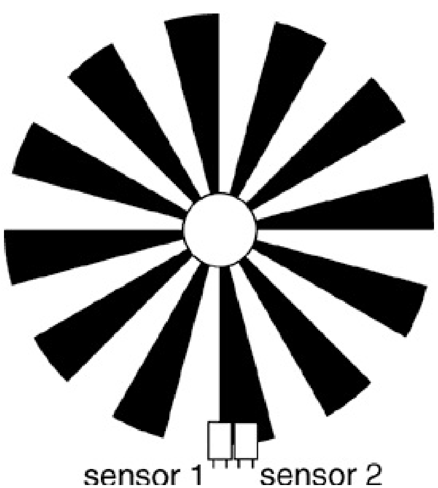
Topics

- Driving MOSFETs and motor (conclusion)
- Battery Model
- Power wiring
- Quiz 1
- Latchup gotcha
- Speed sensing
- Line sensor

Quadrature Encoder



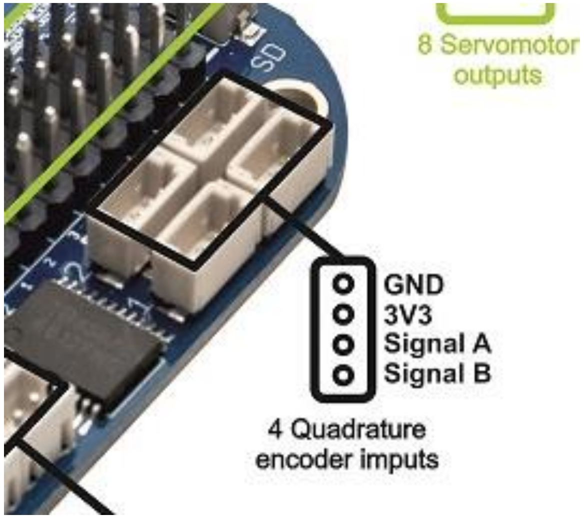
https://www.sinotech.com/wp-content/uploads/quadrature-encoder.gif



Fab suggestion: aluminum foil covered with black paper with slots. 4 slots probably enough. Note: sensors can be placed where convenient- don’t need to look at same slot.

http://codeforfree.weebly.com/uploads/1/0/1/6/10160088/5858962_orig.png

Beagle Bone Blue Quad Encoder




```
int rc_encoder_read  (int ch)
```

Returns
The current position (signed 32-bit integer)
or -1 and prints an error message is there is a problem.
Ch 1-3 are available
Examples:
rc_test_encoders.c.

Sharp GPS260

Fig.9 Test Circuit for Response Time

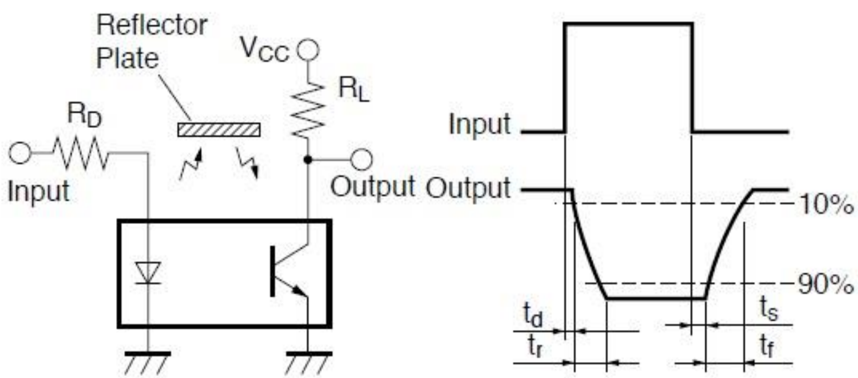
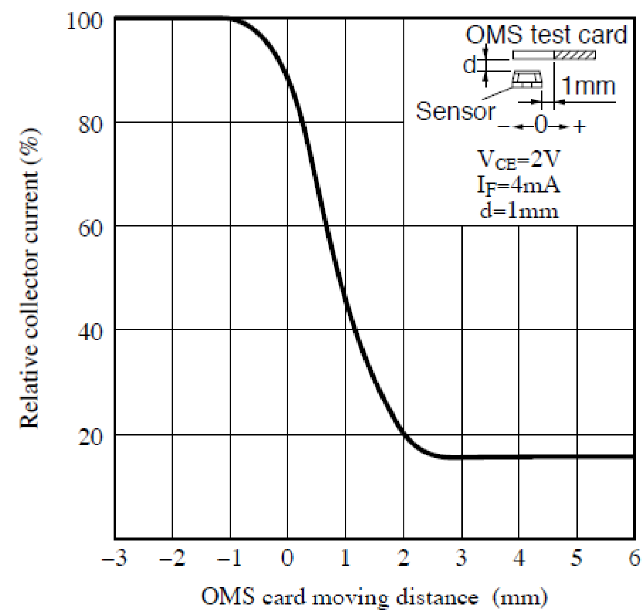
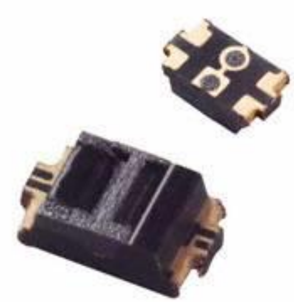


Fig.13 Detecting Position Characteristics (2)



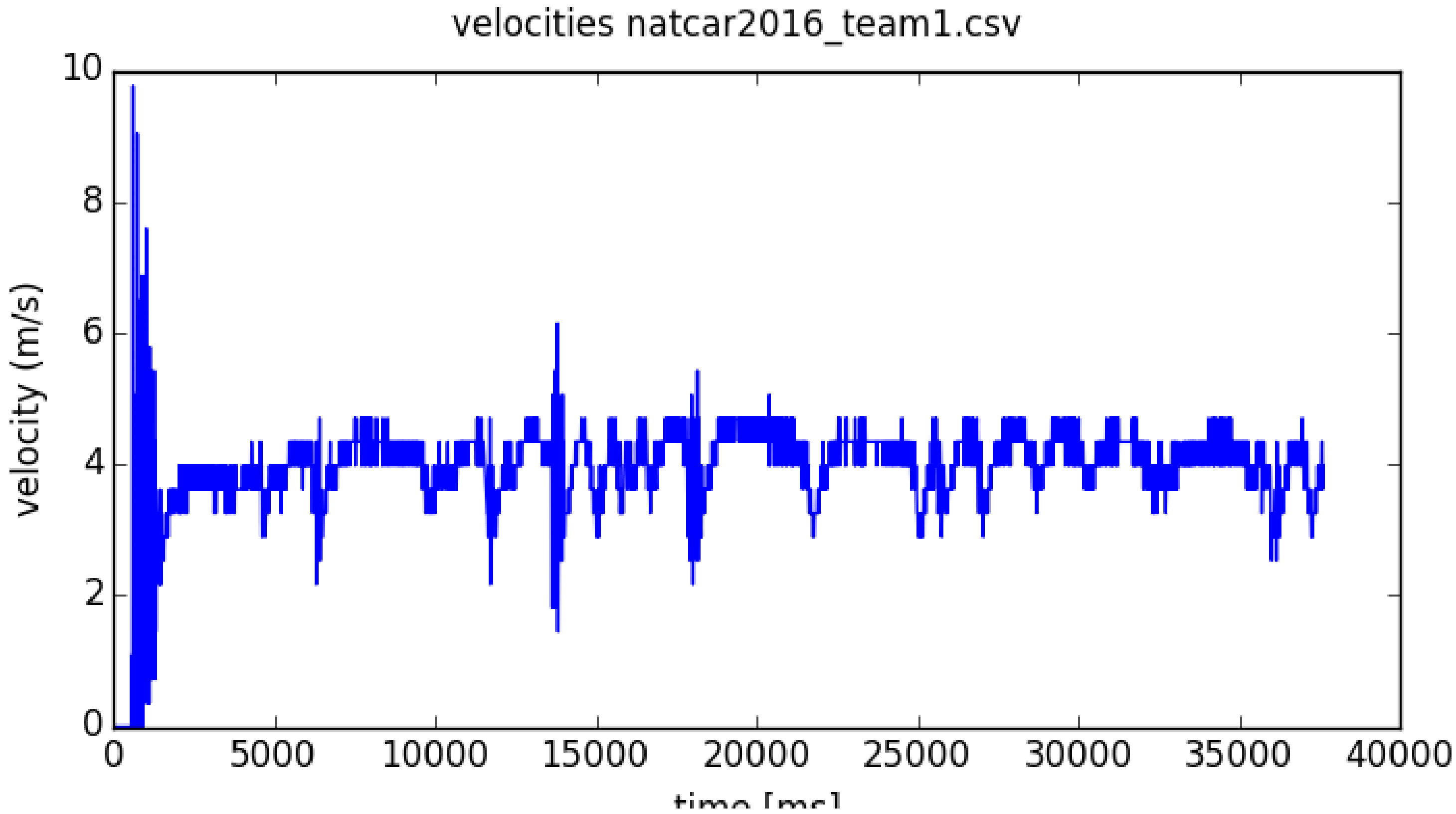
- Choose current 4 mA in LED
- Vcc = 3.3 V
- May want regulated/clean voltage for Vcc



100 us response time

Velocity Sensing

- On board: estimating $\Delta x/\Delta T$

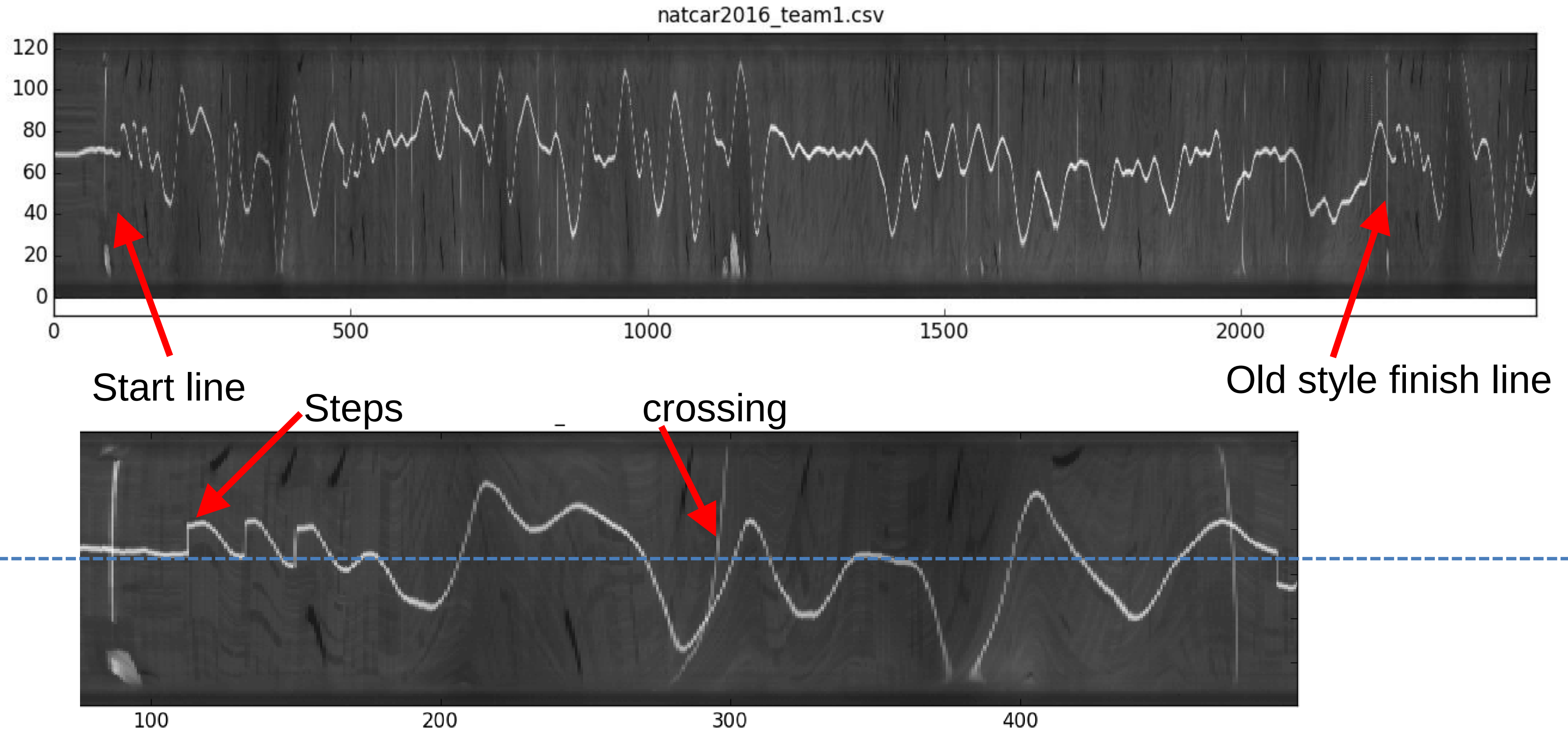


Note: care about velocity sensing usually at cruise speed (also stopping)

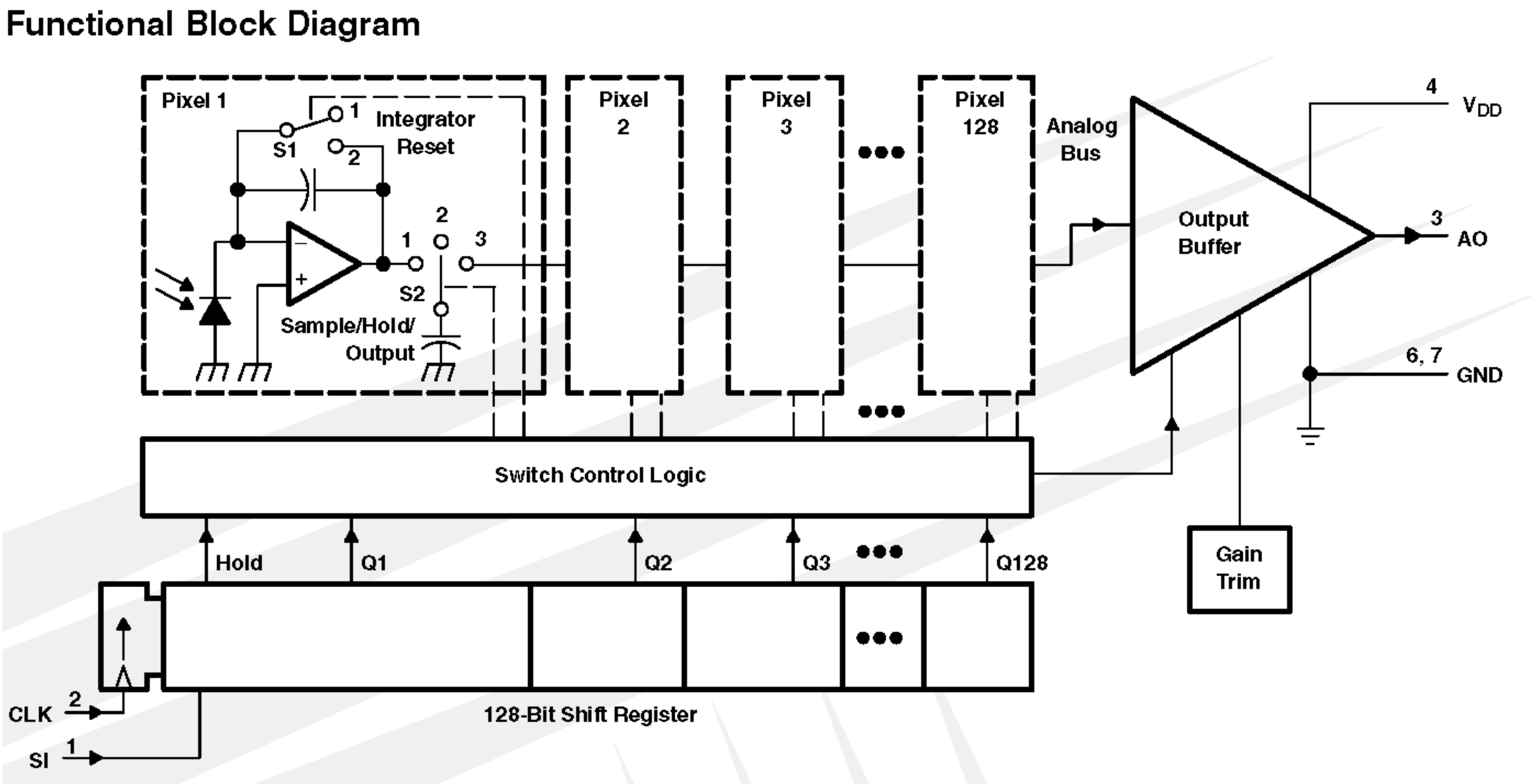
Topics

- Driving MOSFETs and motor (conclusion)
- Battery Model
- Power wiring
- Quiz 1
- Latchup gotcha
- Speed sensing
- Line sensor

Example Line camera data



TSL 1401 line sensor



TSL 1401 line sensor

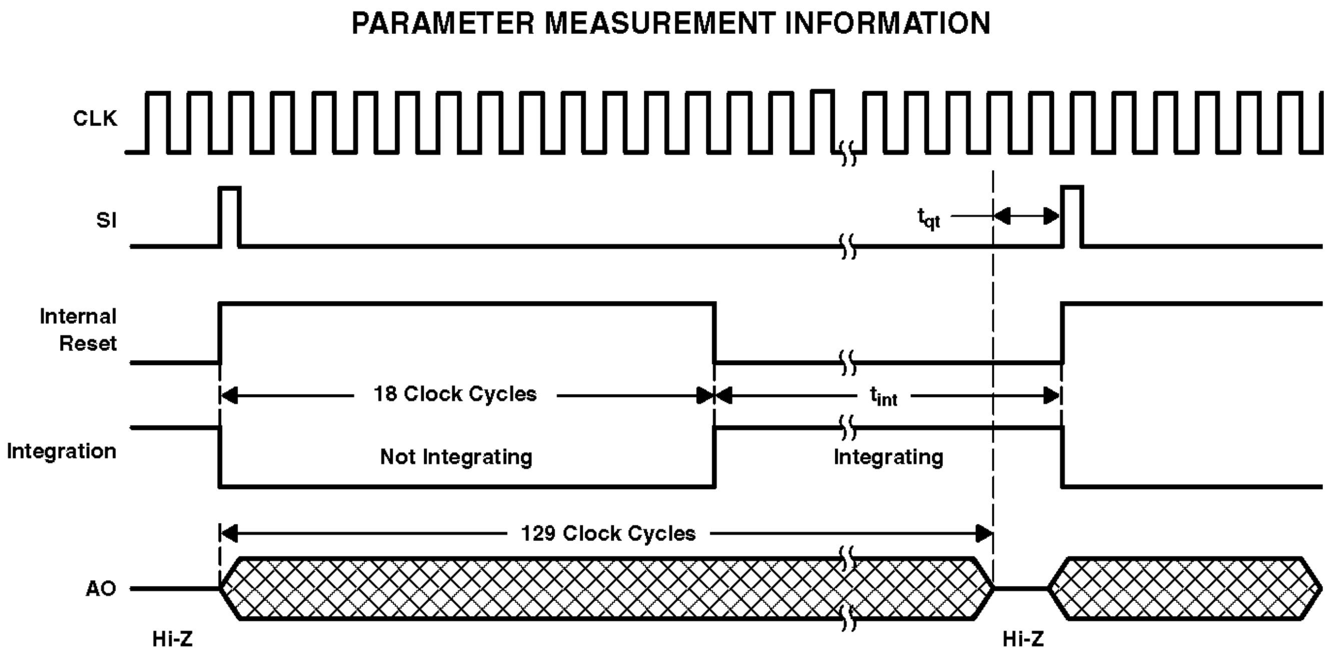


Figure 1. Timing Waveforms

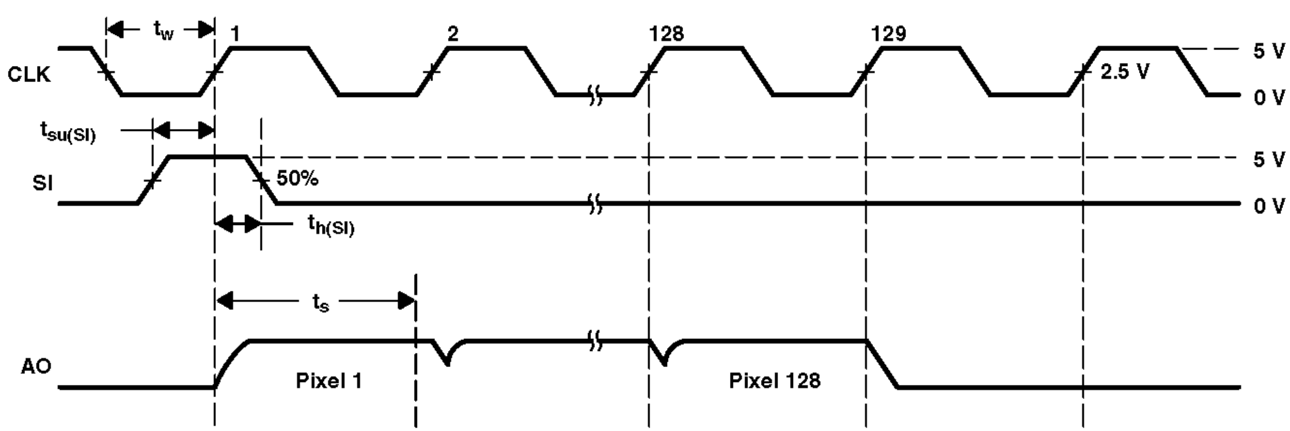


Figure 2. Operational Waveforms

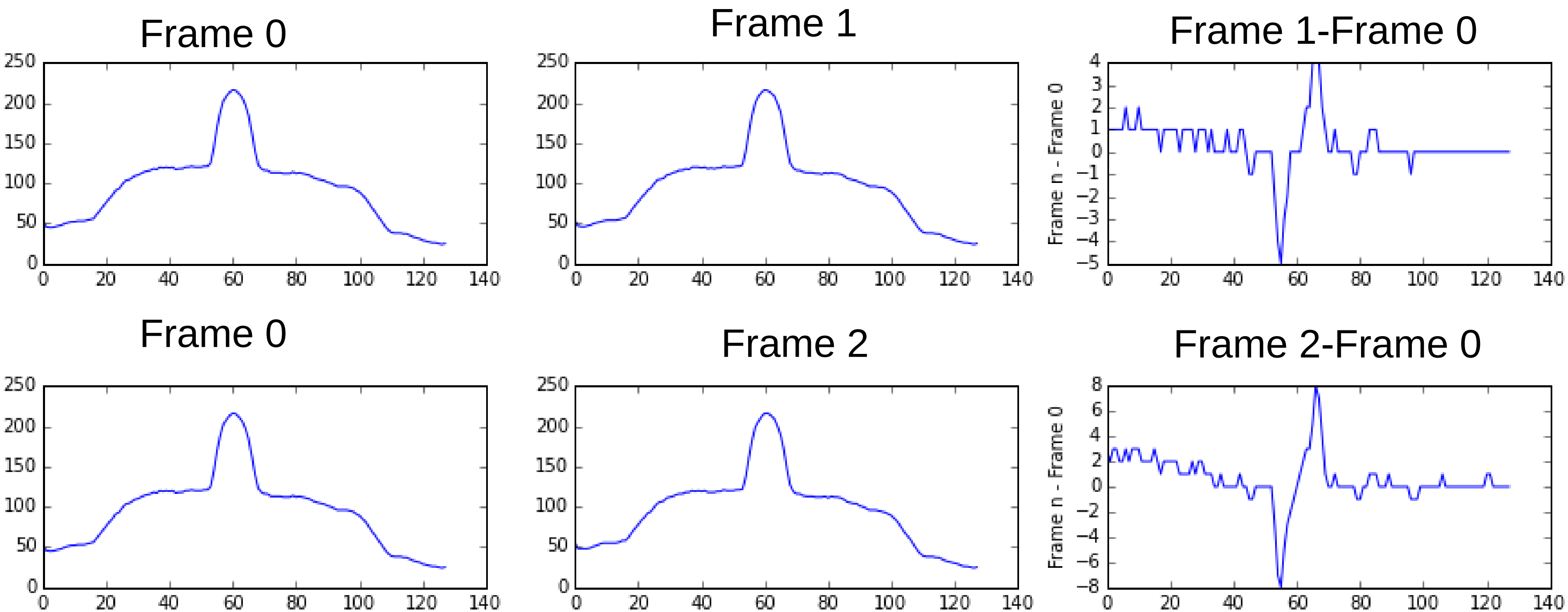
Line sensor

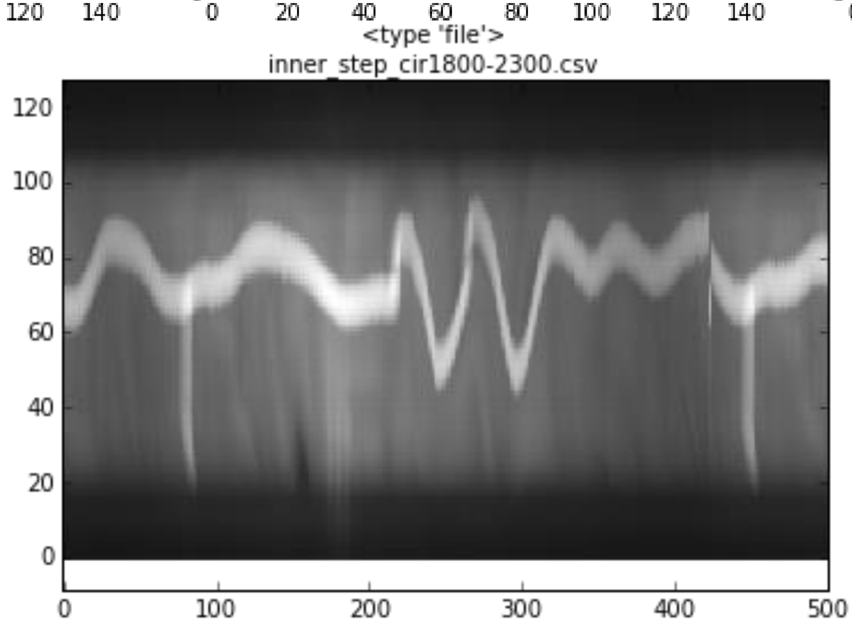
Possible algorithms for line detection

e.g. scipy.signal.filter

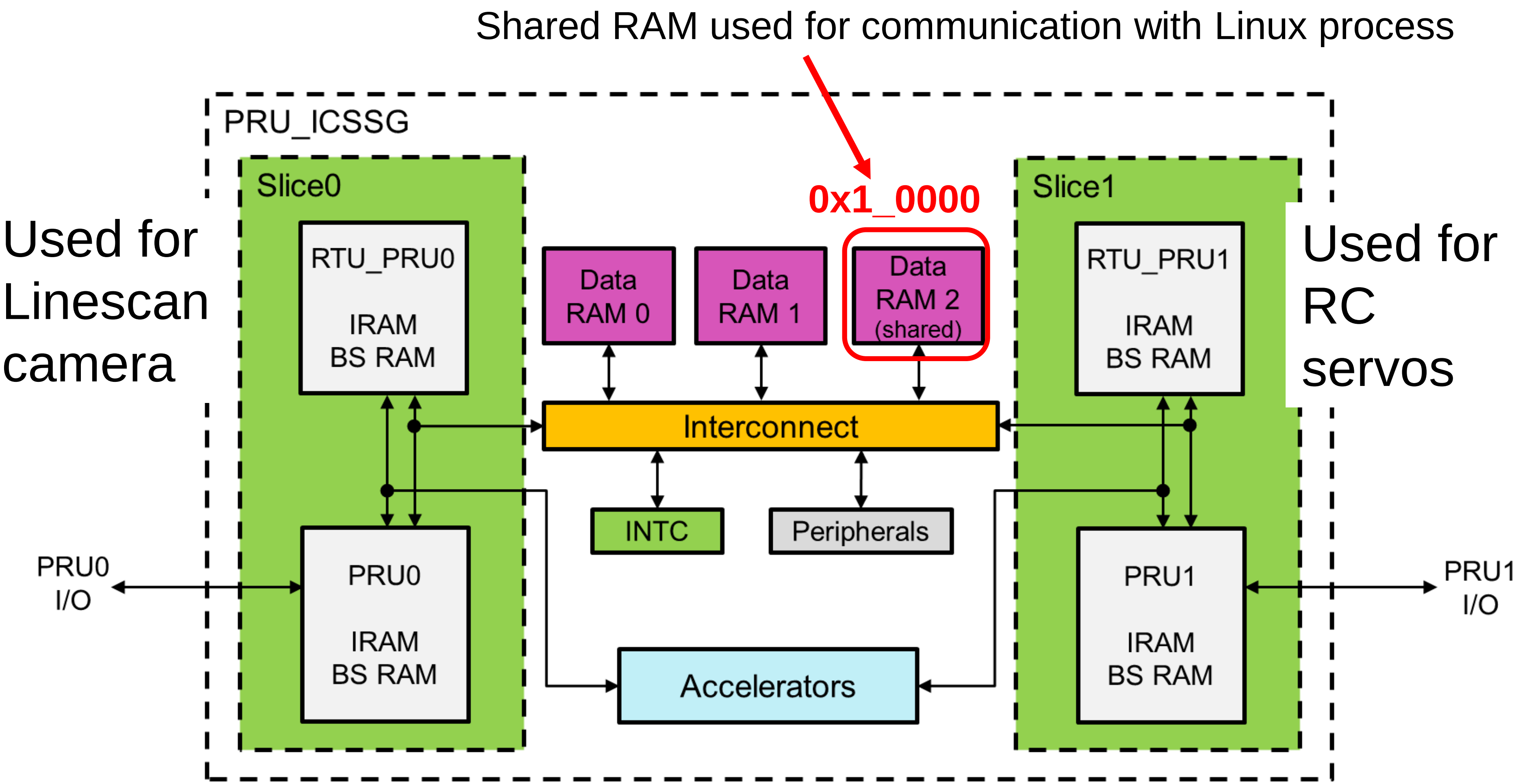
- Subtraction- to find left and right edge of line (ok if not noisy, somewhat lighting invariant)
- Difference of gaussians (idea is to smooth then differentiate)
- Correlation (best match position for known features)
  - scipy.signal.correlate

TSL 1401 line sensor NATCAR 8 bit





Programmable Real-time Unit (PRU)



Used on BeagleBone Blue
PRU1: RC servo
PRU0: real-time A/D for line camera (was originally used for quad encoder Ch4)

Shared 12 kb Address Space

PRU0		**LINUX**
0x1_0000	For encoder	`rc_pru_shared_mem_ptr();`
	Flag = 1 to start conversion	`shared_mem_32bit_ptr[16+1]`
	Pixel 0	`shared_mem_32bit_ptr[16+2]`
	…	
	Pixel 127	`shared_mem_32bit_ptr[16+2+127]`

Linux LineCamera.c line 116:

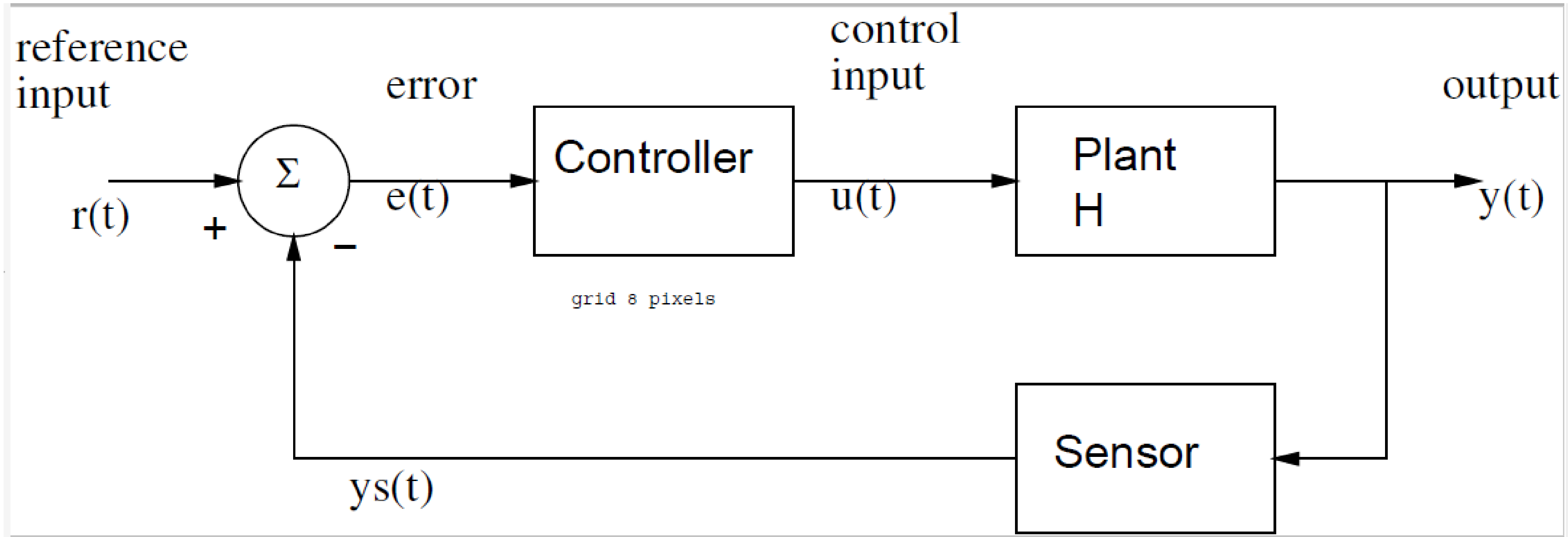
```
shared_mem_32bit_ptr[ENCODER_MEM_OFFSET+1] = 1;
// set flag to start conversion by PRU
while(shared_mem_32bit_ptr[ENCODER_MEM_OFFSET+1] == 1);
```

PRU main_pru0.c line 117:

```
while(shared_mem_32bit_ptr[ENCODER_MEM_OFFSET+1] == 0);
// loop until command
//… read 128 pixels …
shared_mem_32bit_ptr[ENCODER_MEM_OFFSET+1] = 0;
// reset to zero
```

Extra Slides

Velocity control overview



On board…

Proportional control:

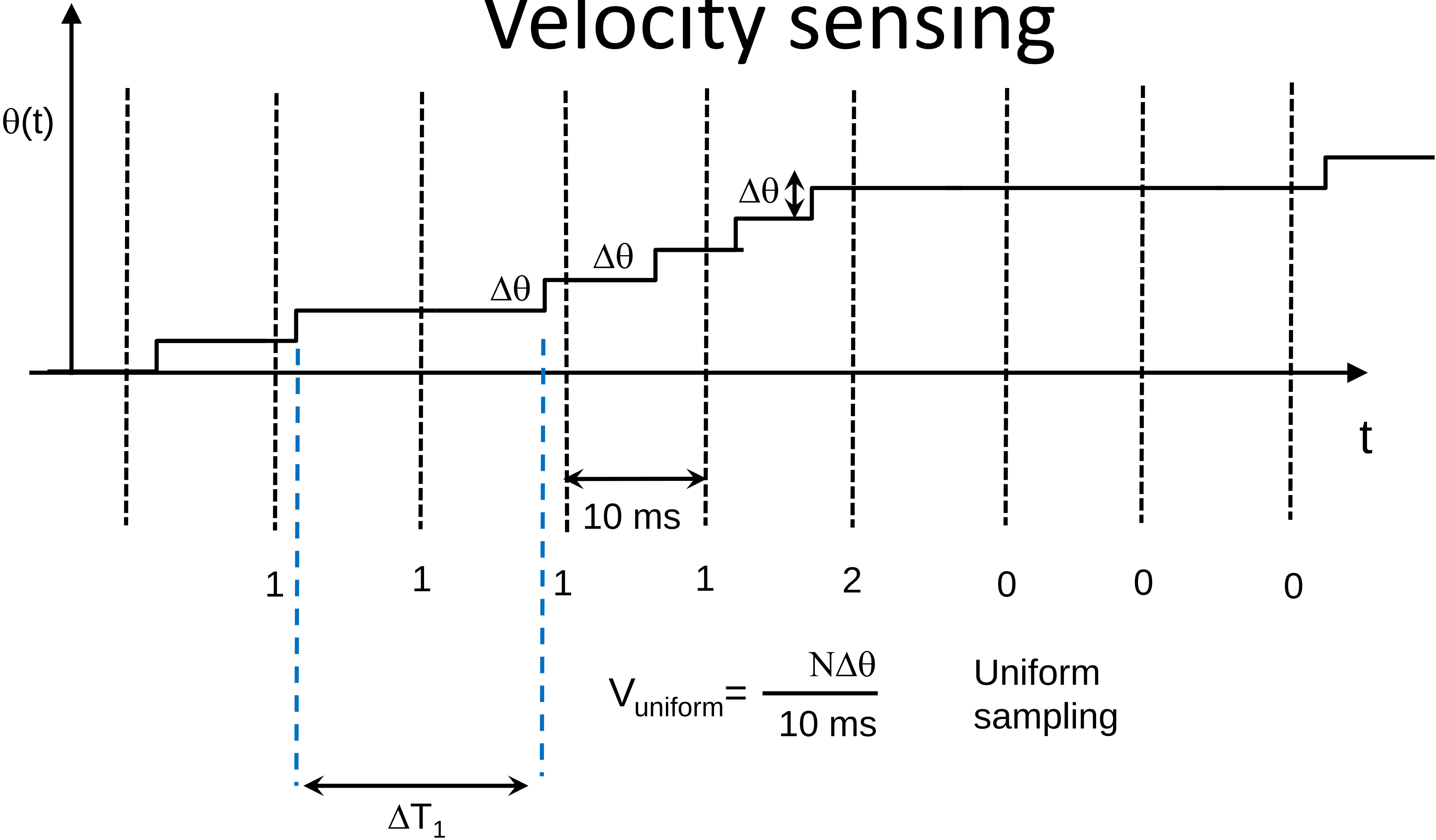
U = kp*e = kp* (r-y);

Here: r is desired velocity, U is PWM %

Proportional + integral control

```
U = kp*e + ki * e_sum;
e_sum = e_sum + e;
```

Velocity sensing



$$V_{uniform} = \frac{N\Delta\theta}{10\text{ ms}}$$

Uniform sampling

$$V_{edge} = \frac{\Delta\theta}{\Delta T_1}$$

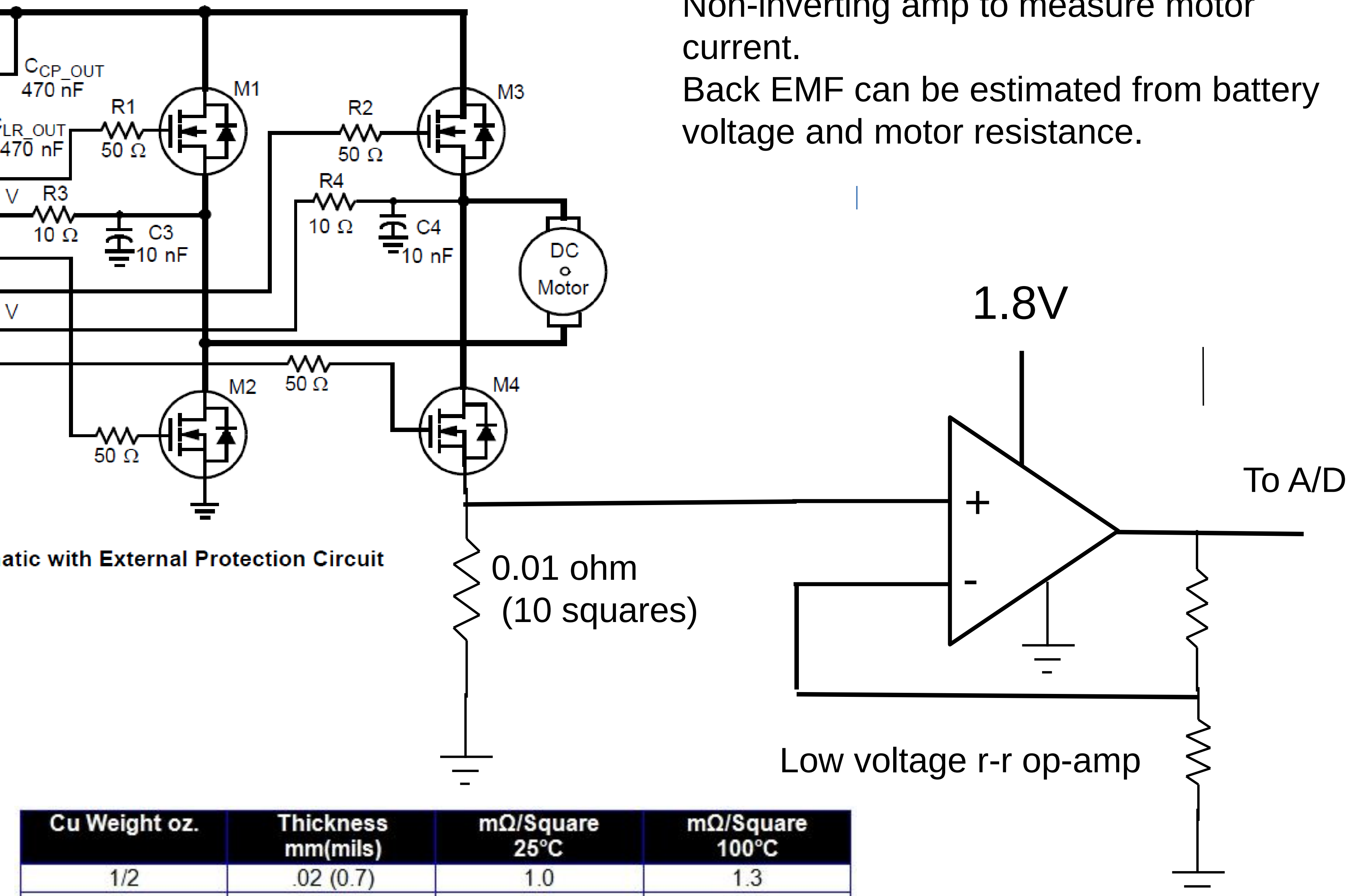
Edge timing

Velocity sensing (recap)

V~ (change in angle)/(change in time)

On board…

Non-inverting amp to measure motor current.
Back EMF can be estimated from battery voltage and motor resistance.

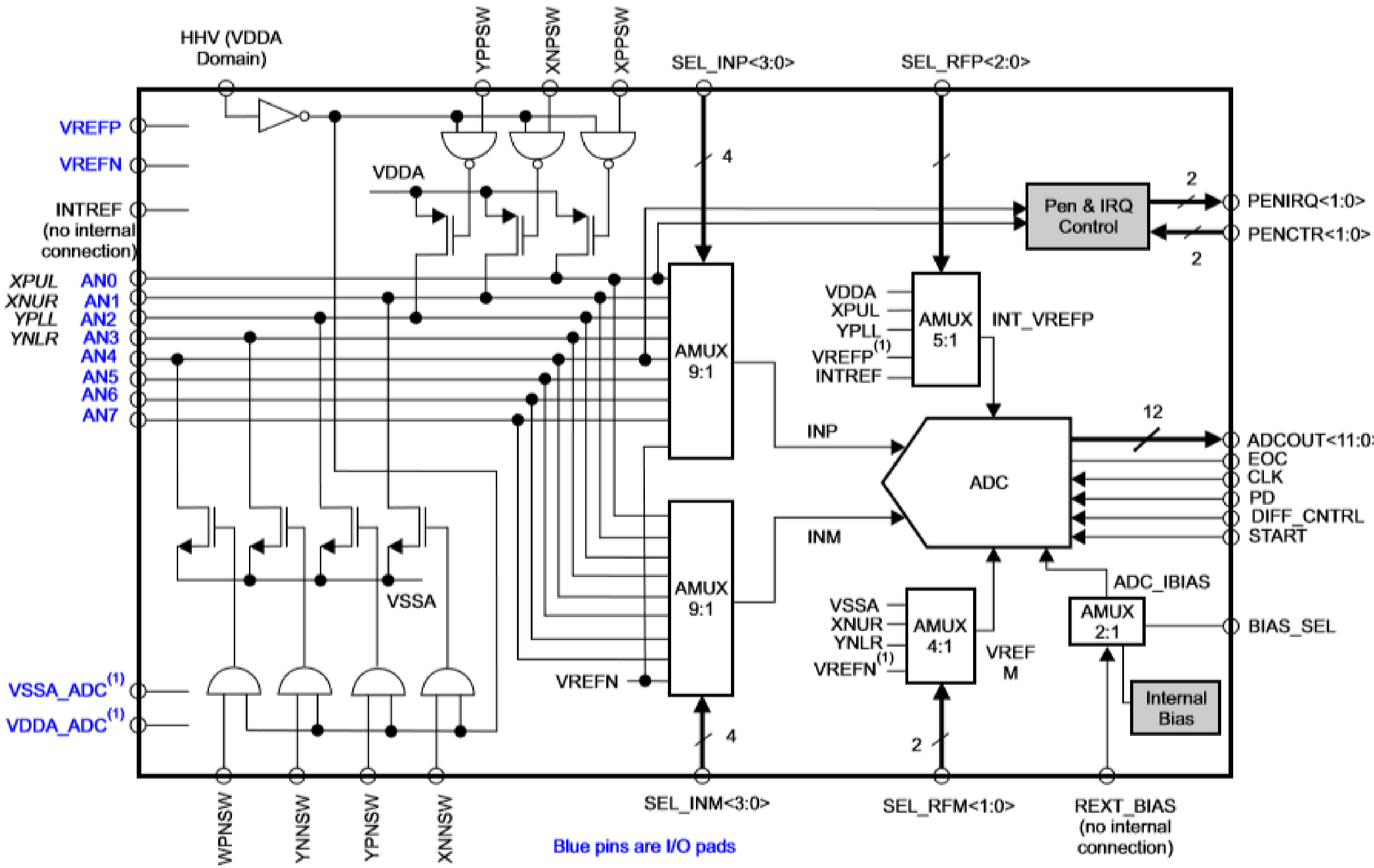


Cu Weight oz.	Thickness mm(mils)	mΩ/Square 25°C	mΩ/Square 100°C
1/2	.02 (0.7)	1.0	1.3
1	.04 (1.4)	0.5	0.65

Back EMF velocity sensing

Analog/Digital Overview

Figure 12-2. Functional Block Diagram



(1) In the device-specific datasheet:

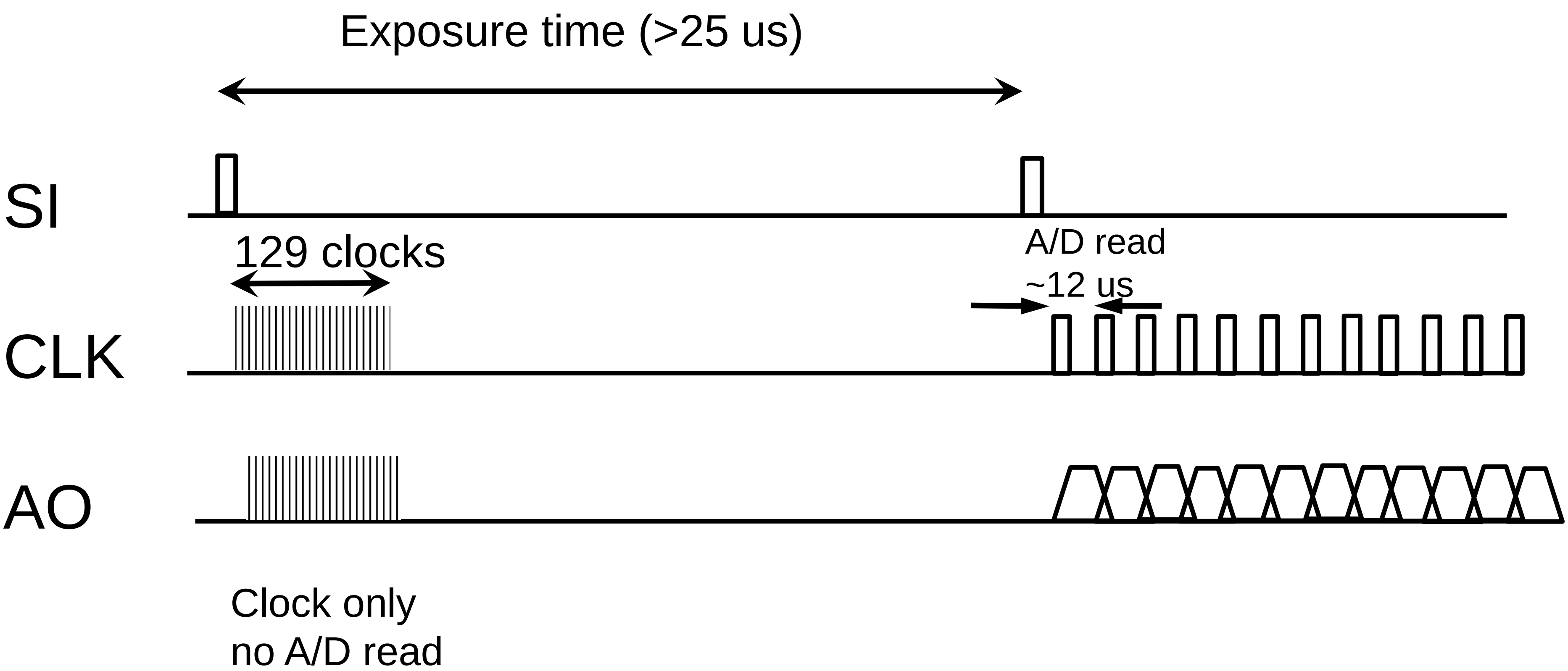
- VDDA_ADC and VSSA_ADC are referred to as "Internal References"
- VREFP and VREFN are referred to as "External References"

Caution: 1.8V MAXIMUM

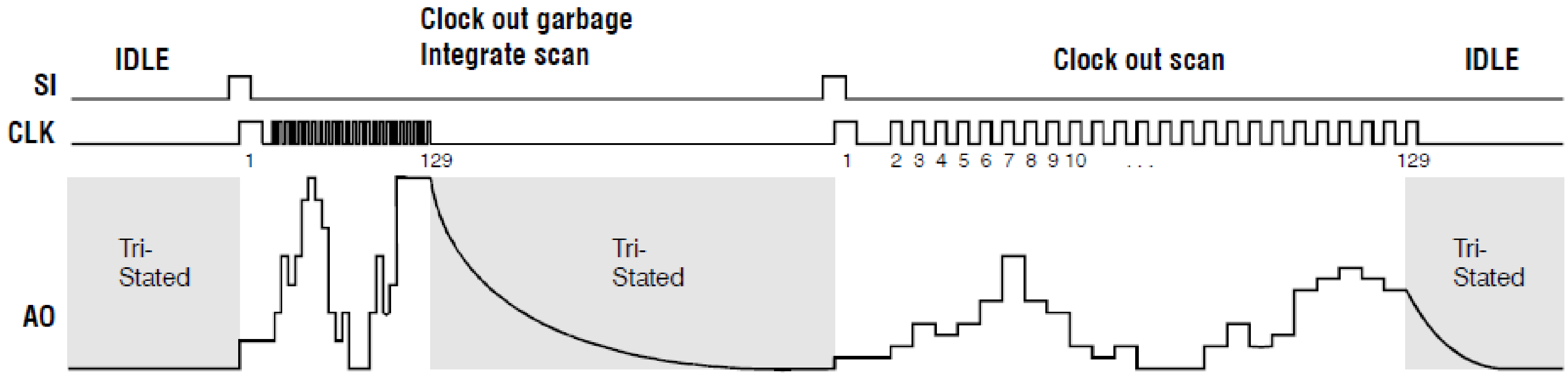
Note: lots of sequencing/delays.

Set up in PRU0 to read 128 elements at ~8 us/sample.

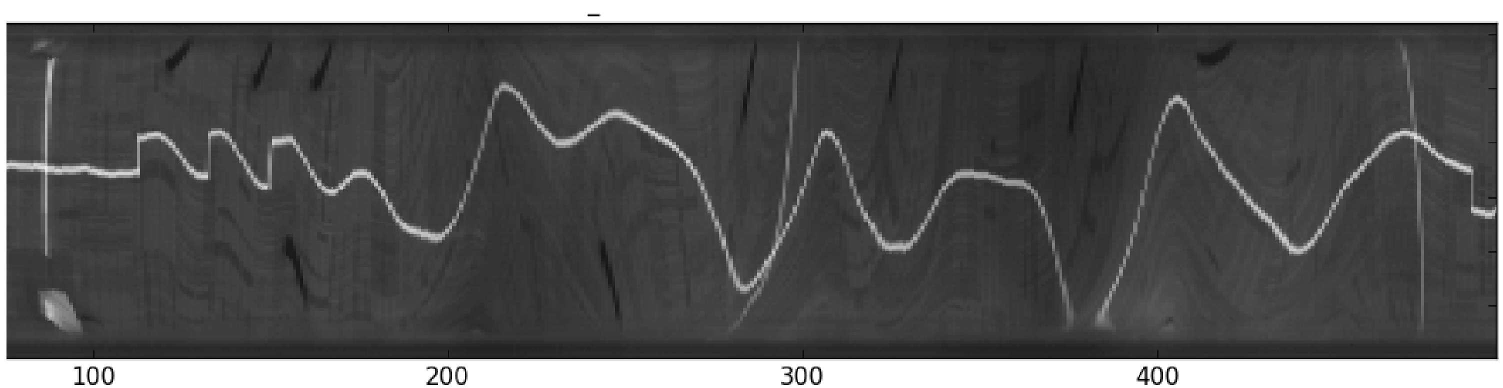
TSL 1401 line sensor- option exposure control
(would need to modify PRU code)

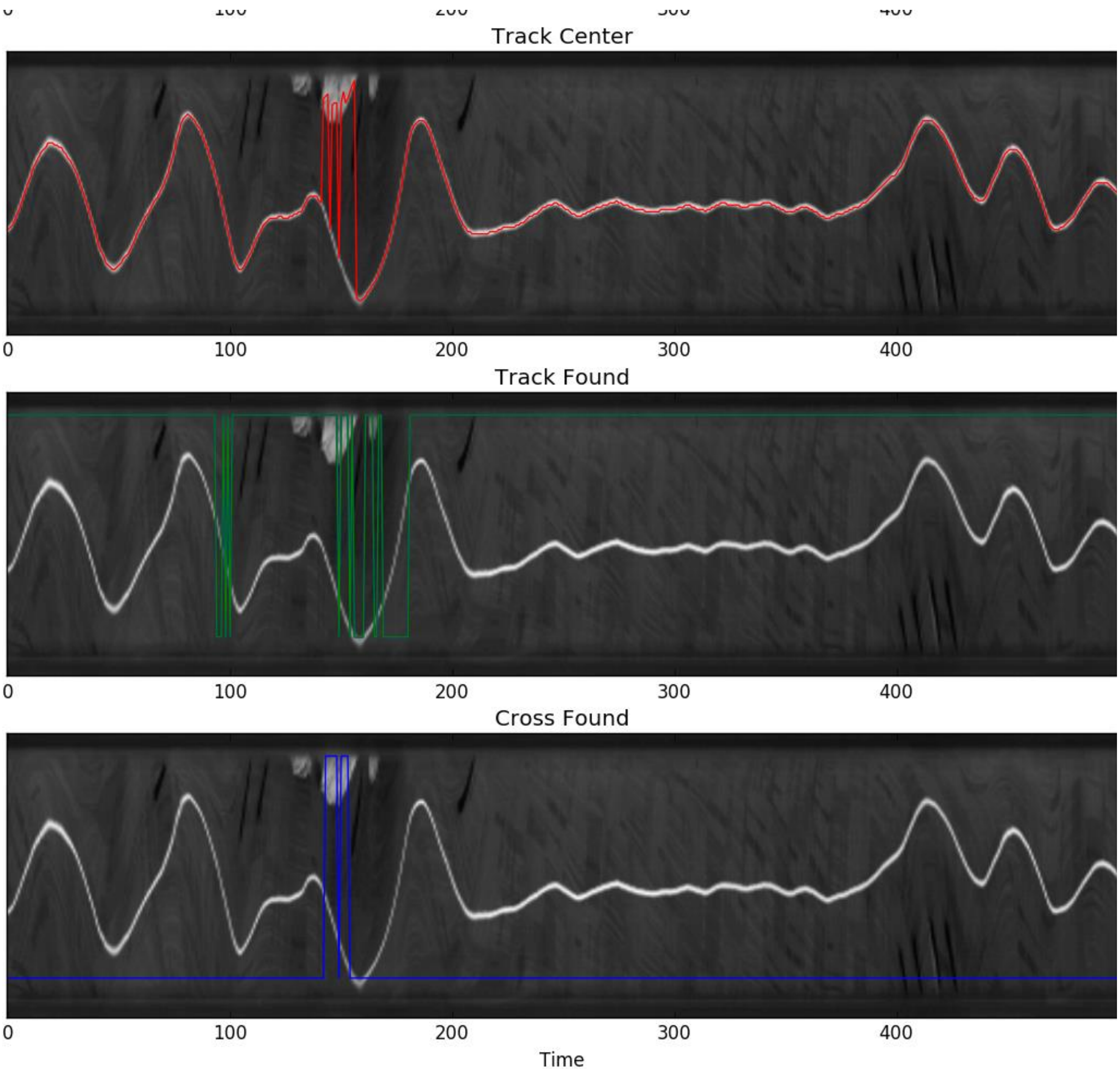


Automatic Gain Control



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




0
100
200
300
400
Track Center
0
100
200
300
400
Track Found
0
100
200
300
400
Cross Found
0
100
200
300
400
Time