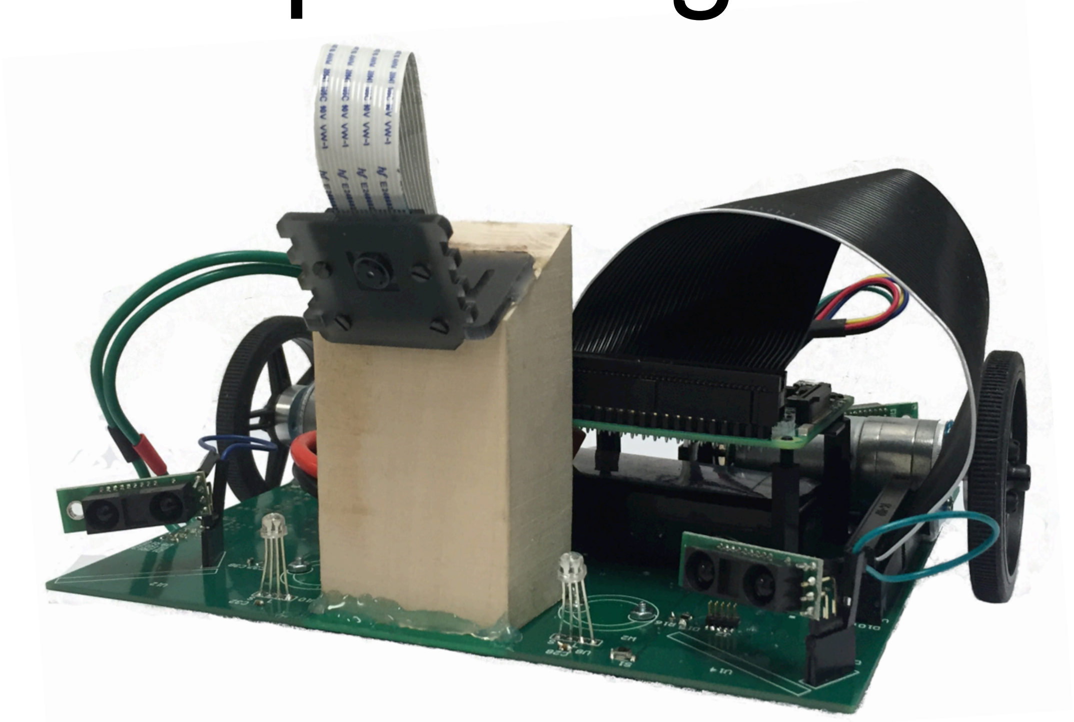
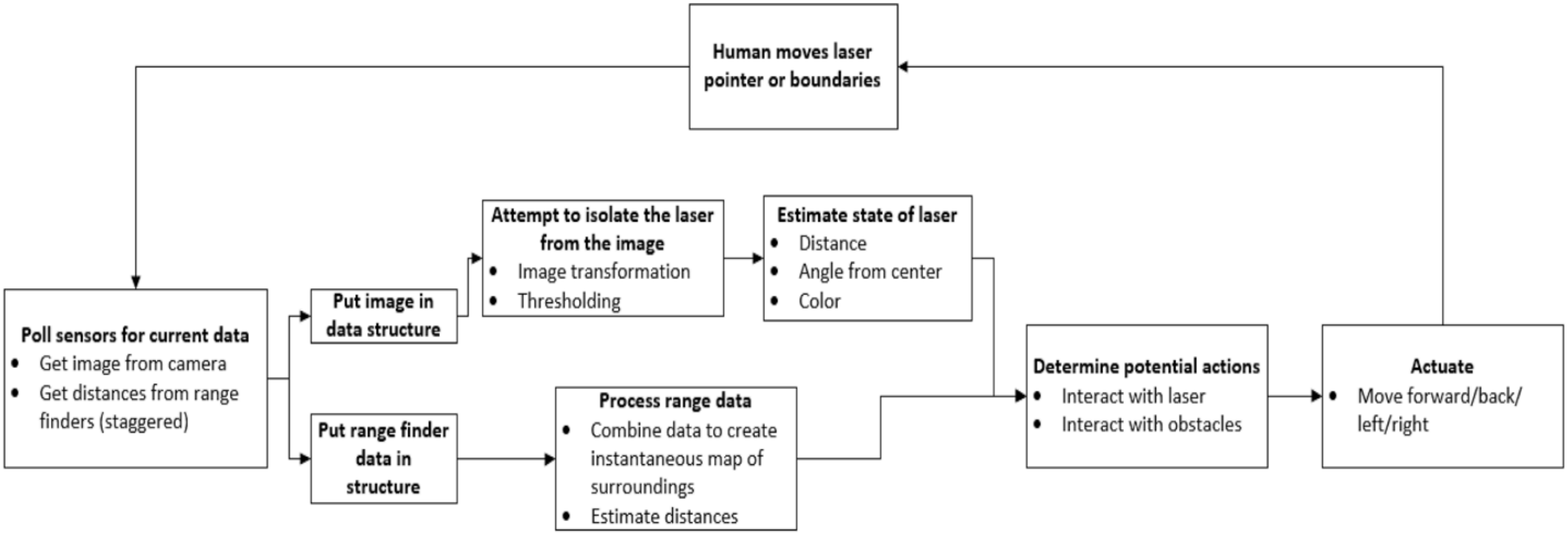
DOGBOT: A Robot Responding to Laser Stimulus



Andrew Levin
Shiva Mehta
Jonathan Zarger

High-Level Overview



Human moves laser pointer or boundaries

Poll sensors for current data
- Get image from camera
- Get distances from range finders (staggered)

Put image in data structure

Attempt to isolate the laser from the image
- Image transformation
- Thresholding

Estimate state of laser
- Distance
- Angle from center
- Color

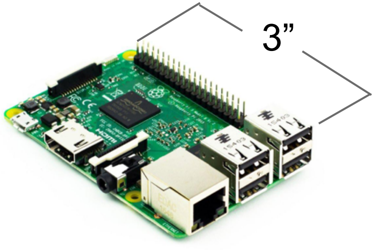
Put range finder data in structure

Process range data
- Combine data to create instantaneous map of surroundings
- Estimate distances

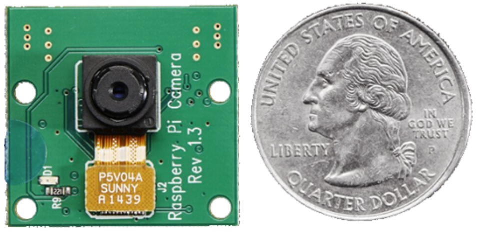
Determine potential actions
- Interact with laser
- Interact with obstacles

Actuate
- Move forward/back/left/right

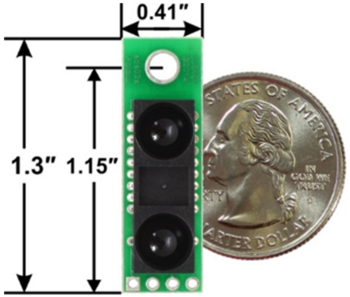
Hardware



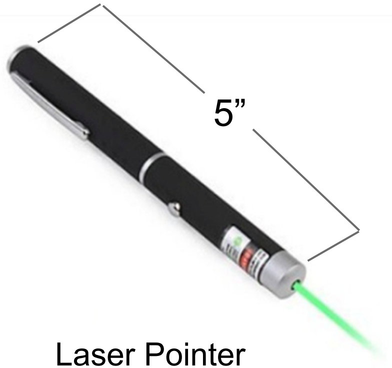
Raspberry Pi 3



Raspberry Pi Camera



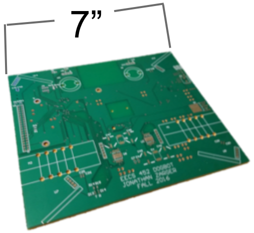
IR Sensor



Laser Pointer

Motors

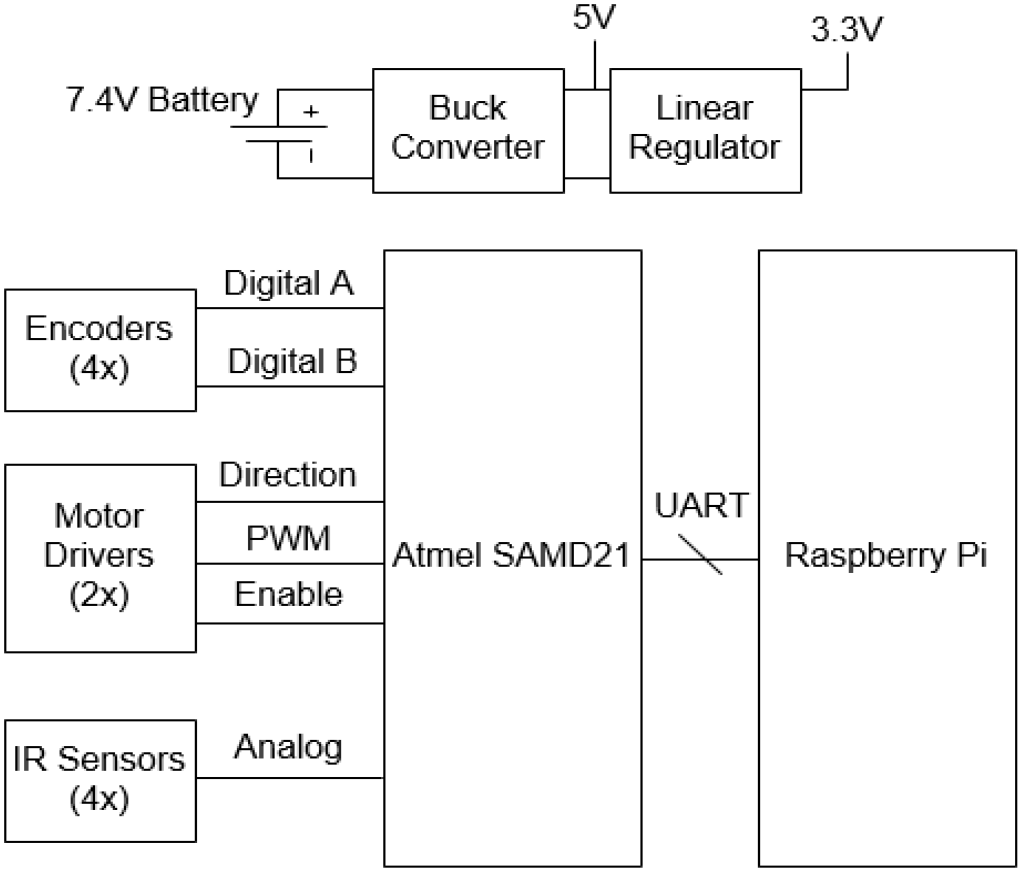
Wheels



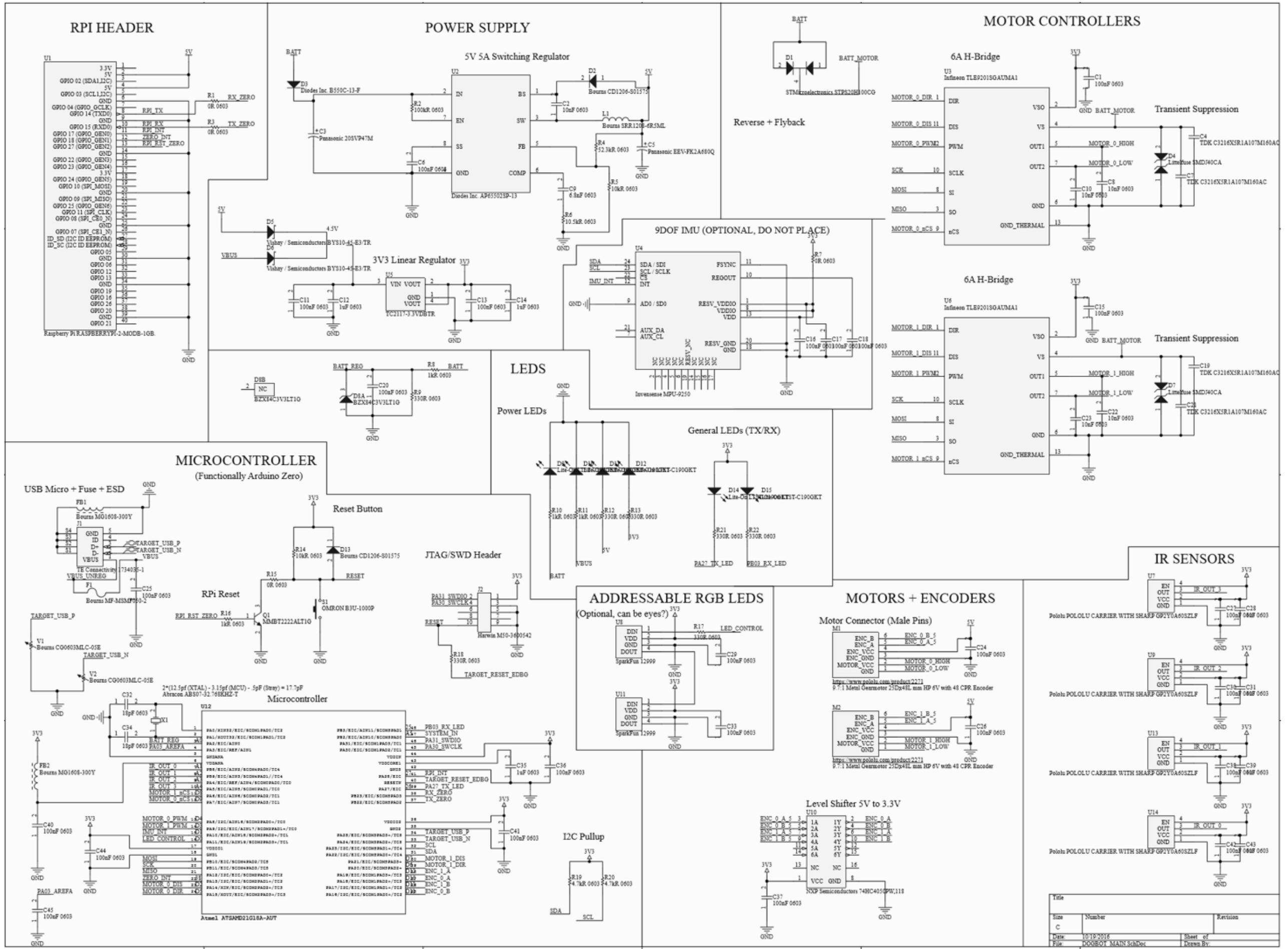
Chassis Board

Misc. Integrated Circuits

Hardware


5V
3.3V
7.4V Battery
+
Buck Converter
Linear Regulator
Digital A
Encoders (4x)
Digital B
Direction
Motor Drivers (2x)
PWM
Enable
UART
Atmel SAMD21
Raspberry Pi
Analog
IR Sensors (4x)

Chassis PCB Schematic Capture



Hardware Challenges and Solutions

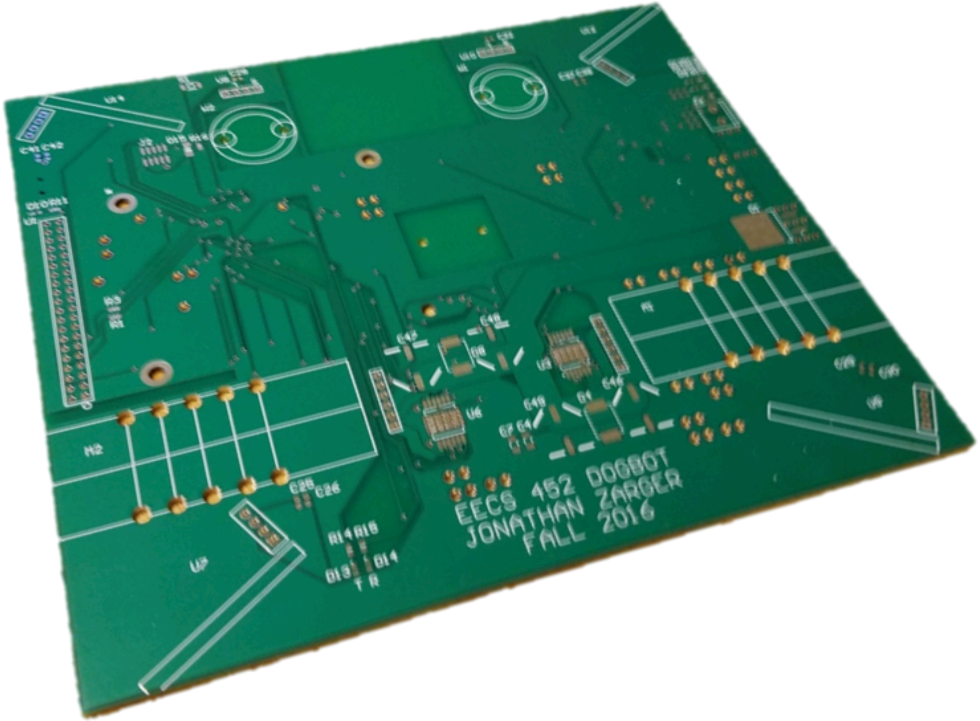
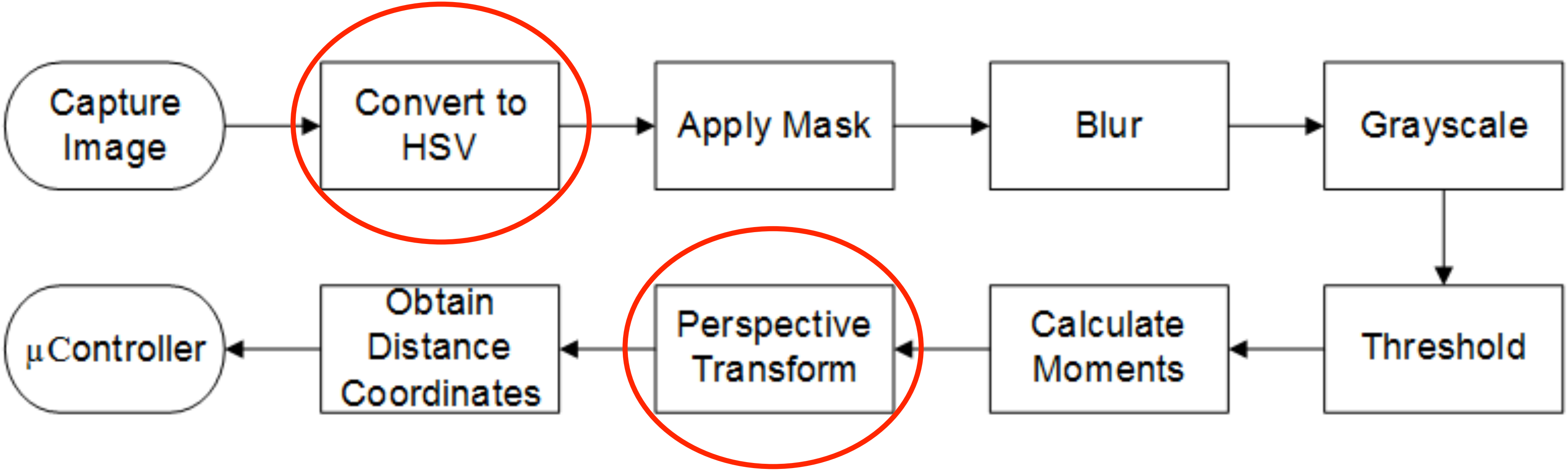


Image Processing Architecture



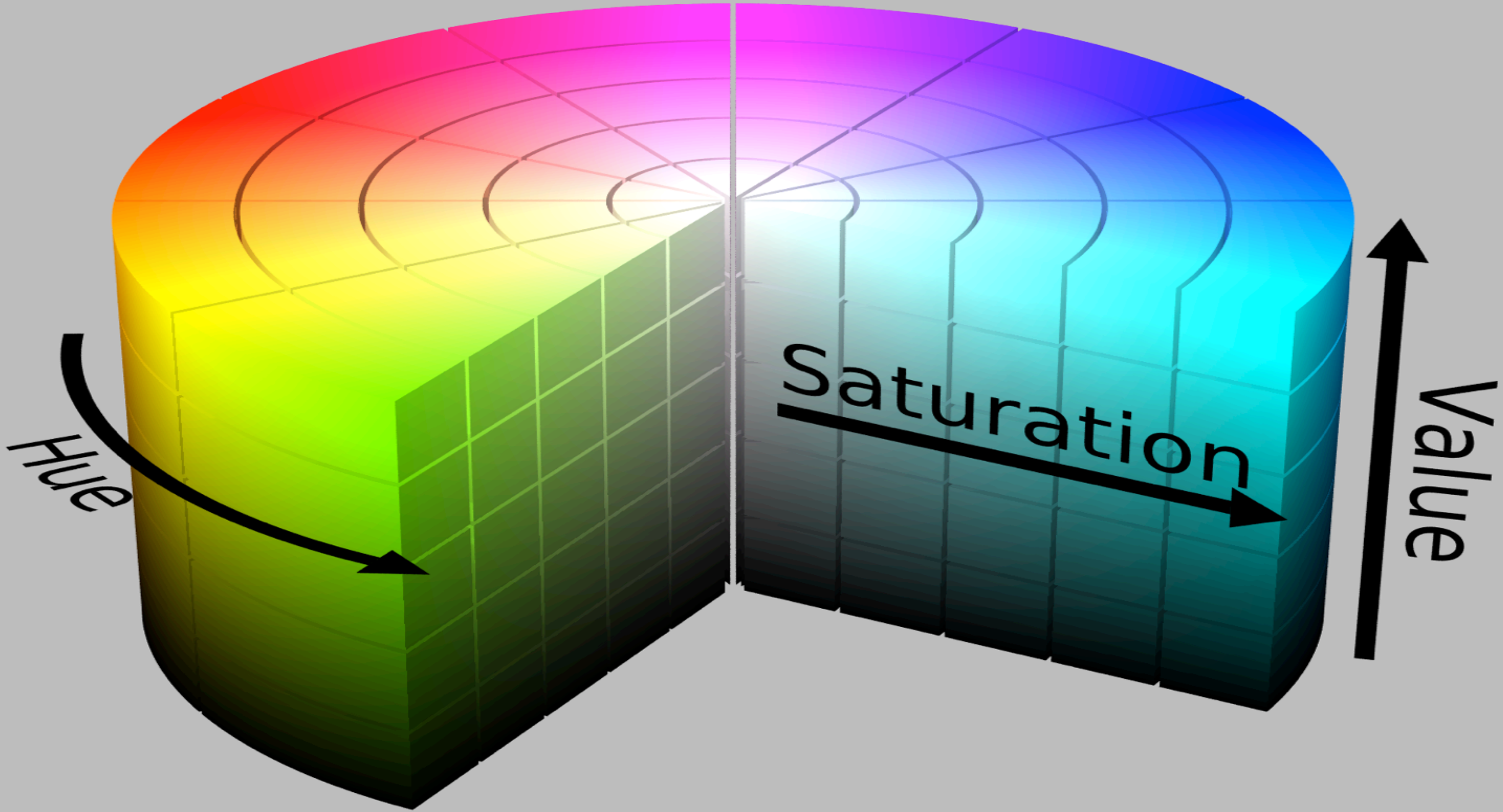
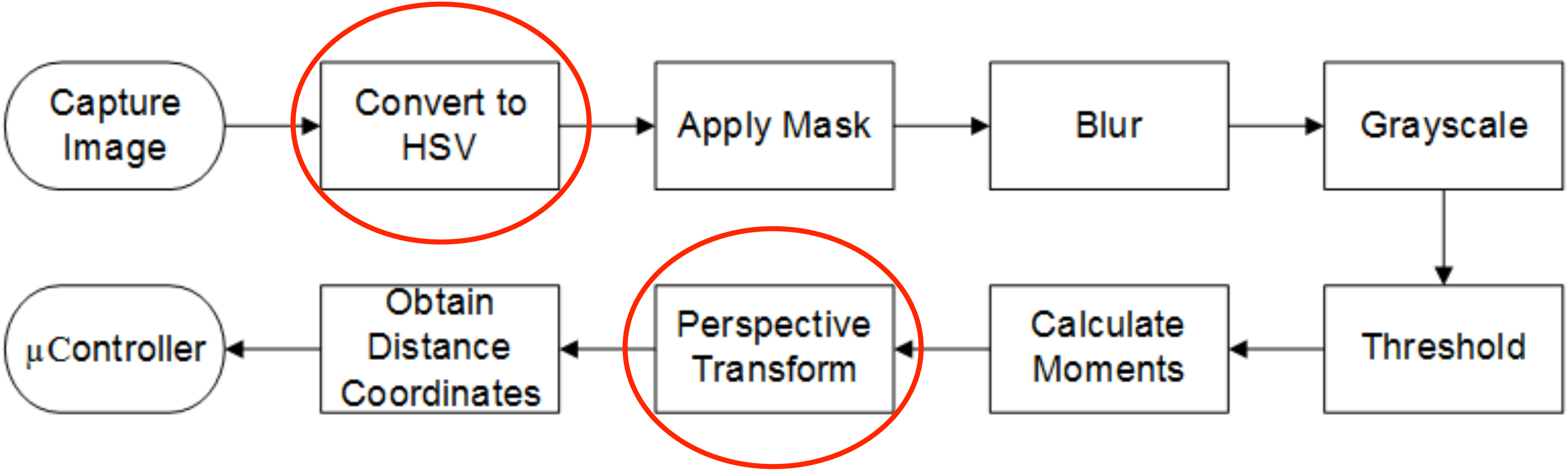
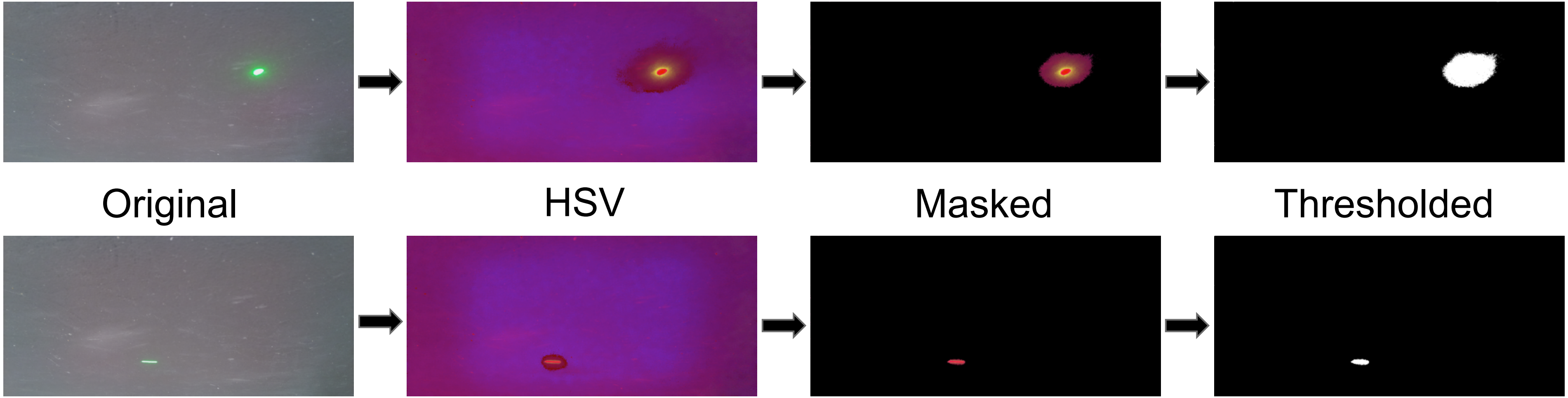

Hue
Saturation
Value

Image Processing Architecture



Through the Eyes of the Robot



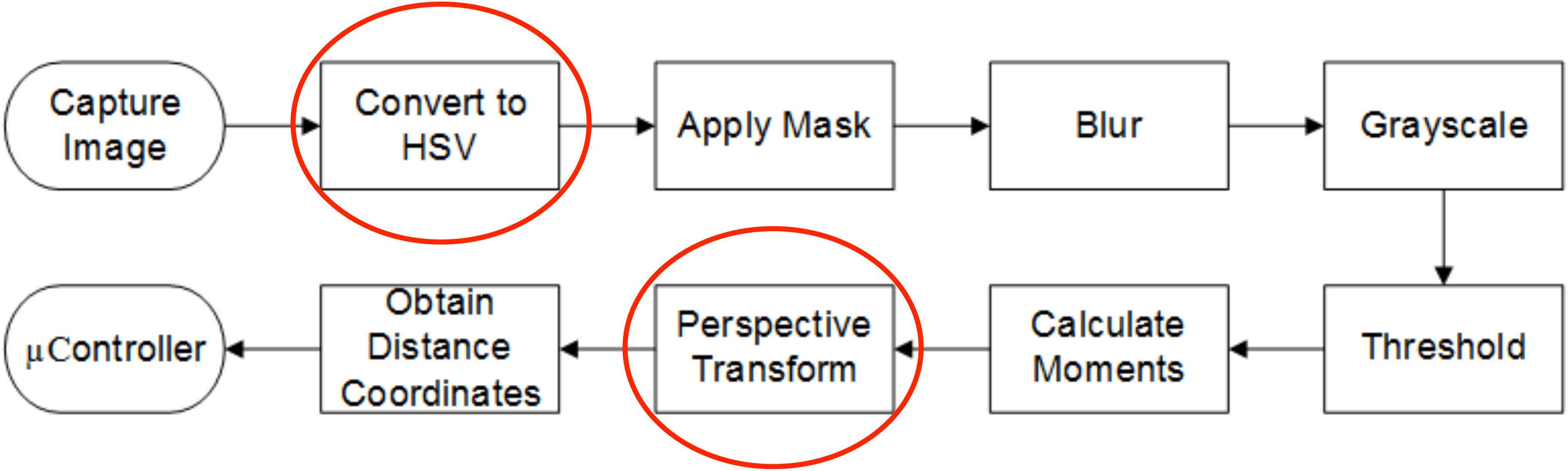
Original

HSV

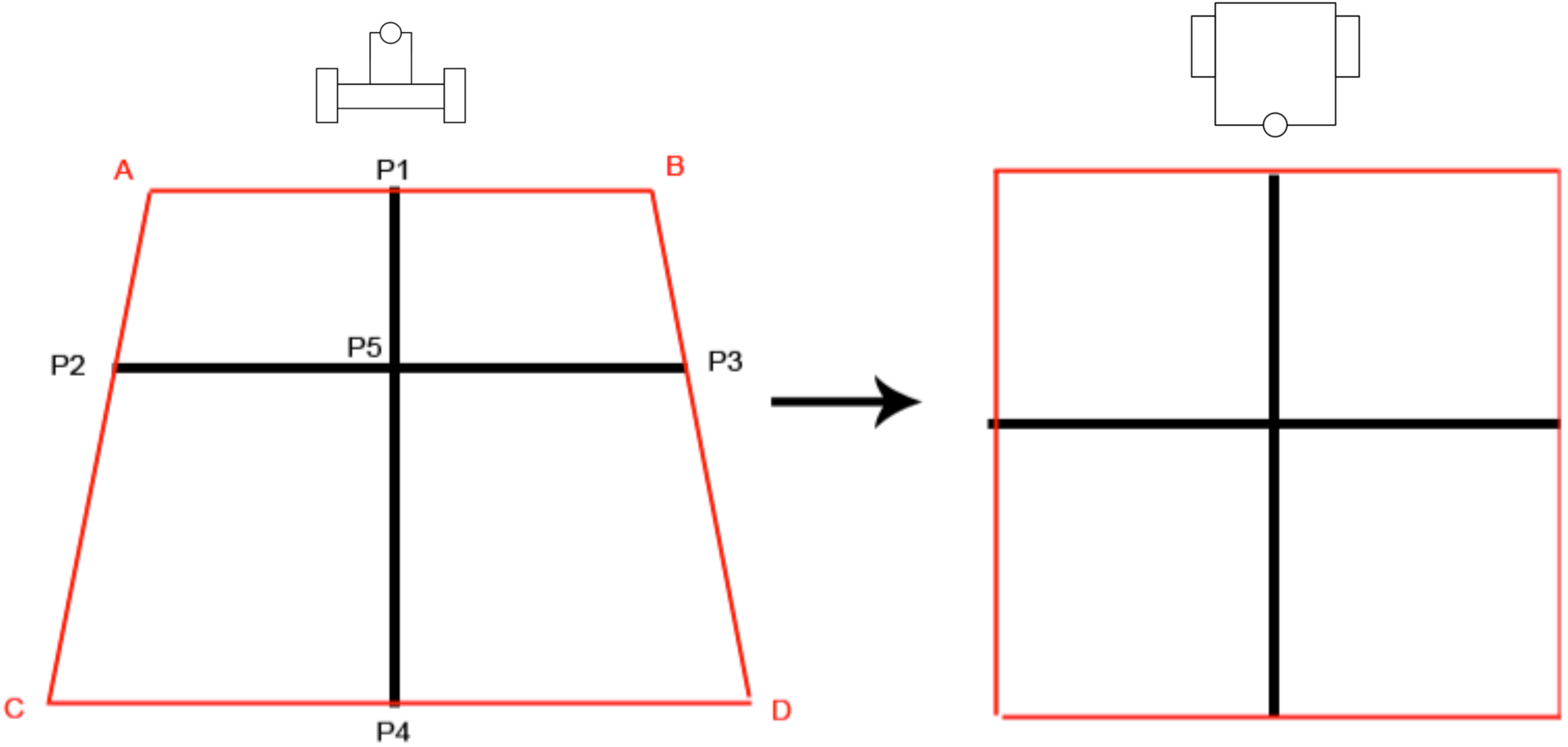
Masked

Thresholded

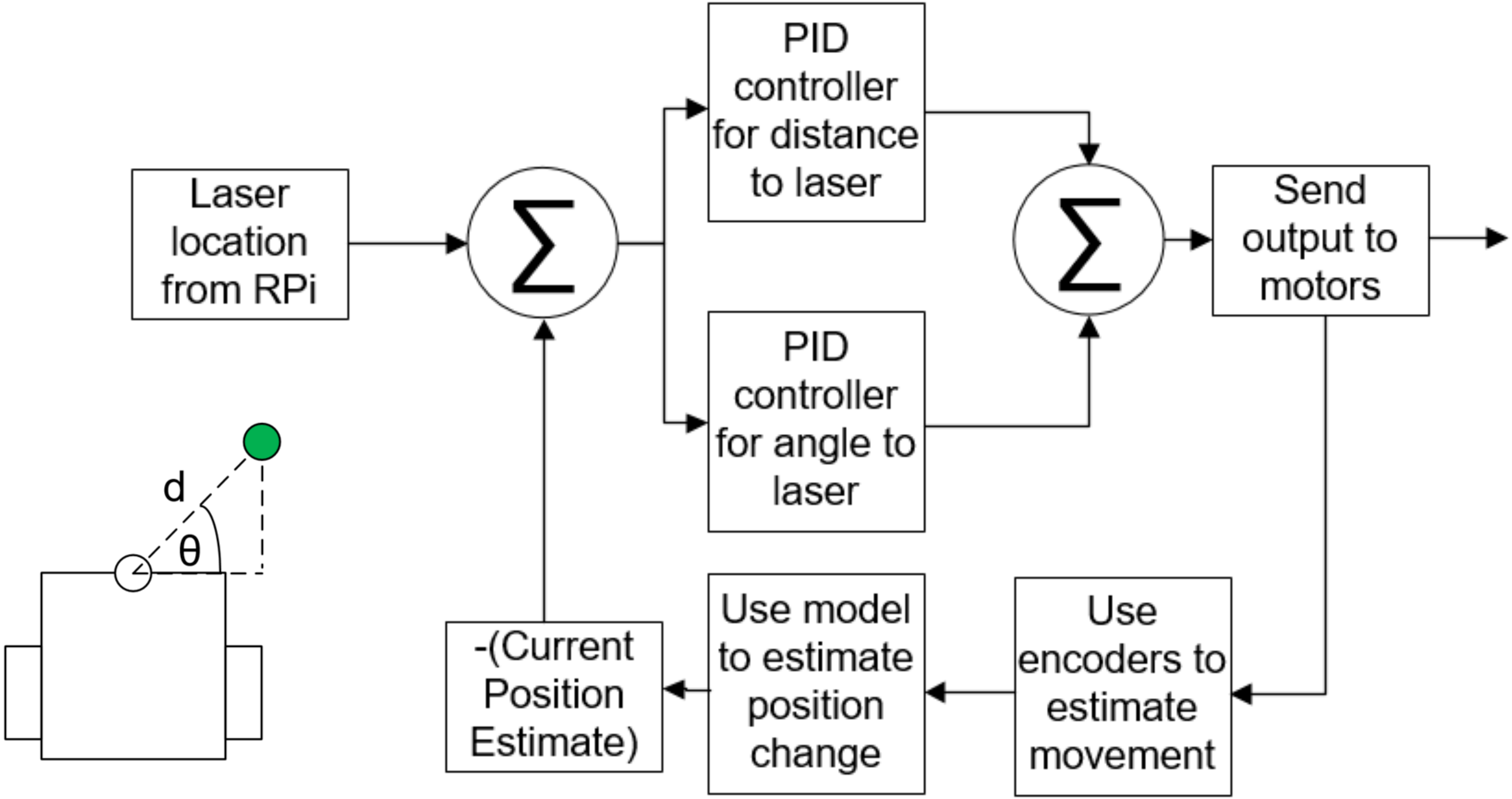
Image Processing Architecture



Perspective Transform



Control System Architecture



Laser location from RPi

Σ

PID controller for distance to laser

PID controller for angle to laser

Σ

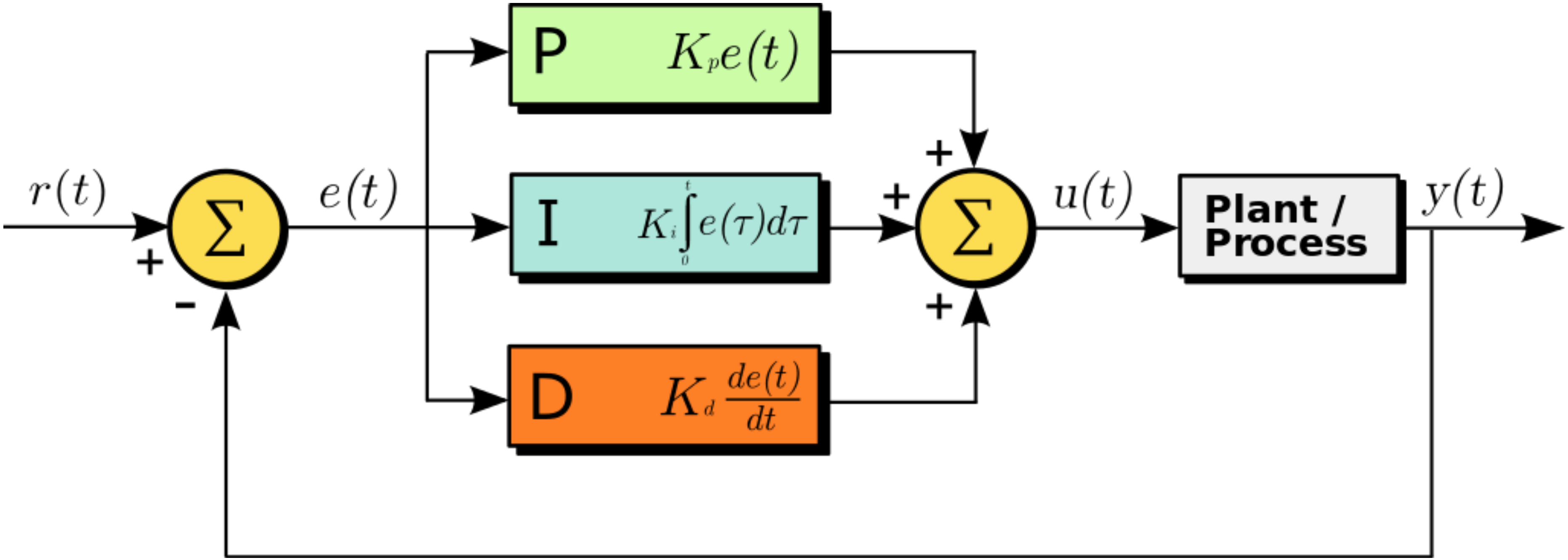
Send output to motors

Use encoders to estimate movement

Use model to estimate position change

-(Current Position Estimate)

PID Controller



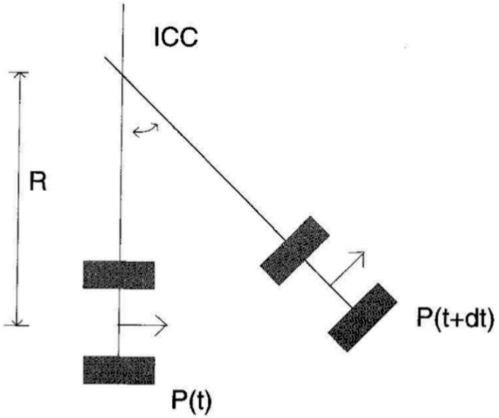
https://upload.wikimedia.org/wikipedia/commons/thumb/4/43/PID_en.svg/971px-PID_en.svg.png

Differential Drive Robot Model

$$R = \frac{l}{2}\frac{V_l + V_r}{V_r - V_l}; \; \omega = \frac{V_r - V_l}{l};$$

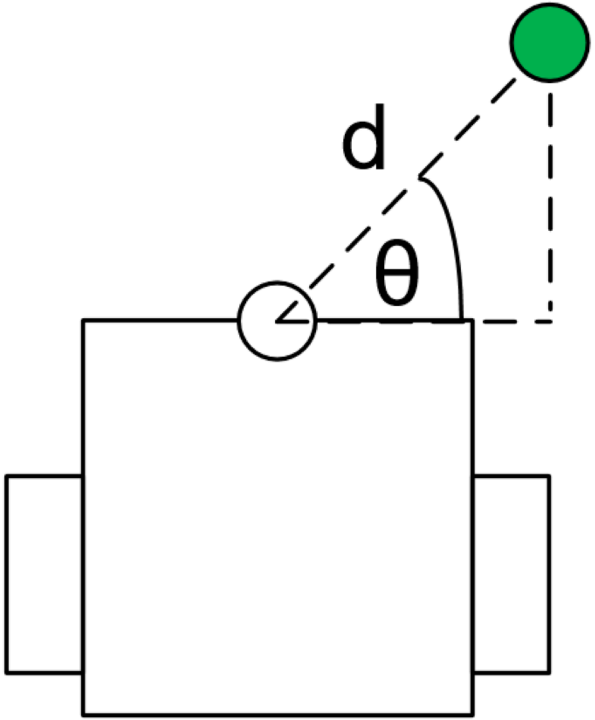
$$ICC = [x - R\,sin(\theta)\,,\, y + R\,cos(\theta)]$$

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} cos(\omega\delta t) & -sin(\omega\delta t) & 0 \\ sin(\omega\delta t) & cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\delta t \end{bmatrix}$$



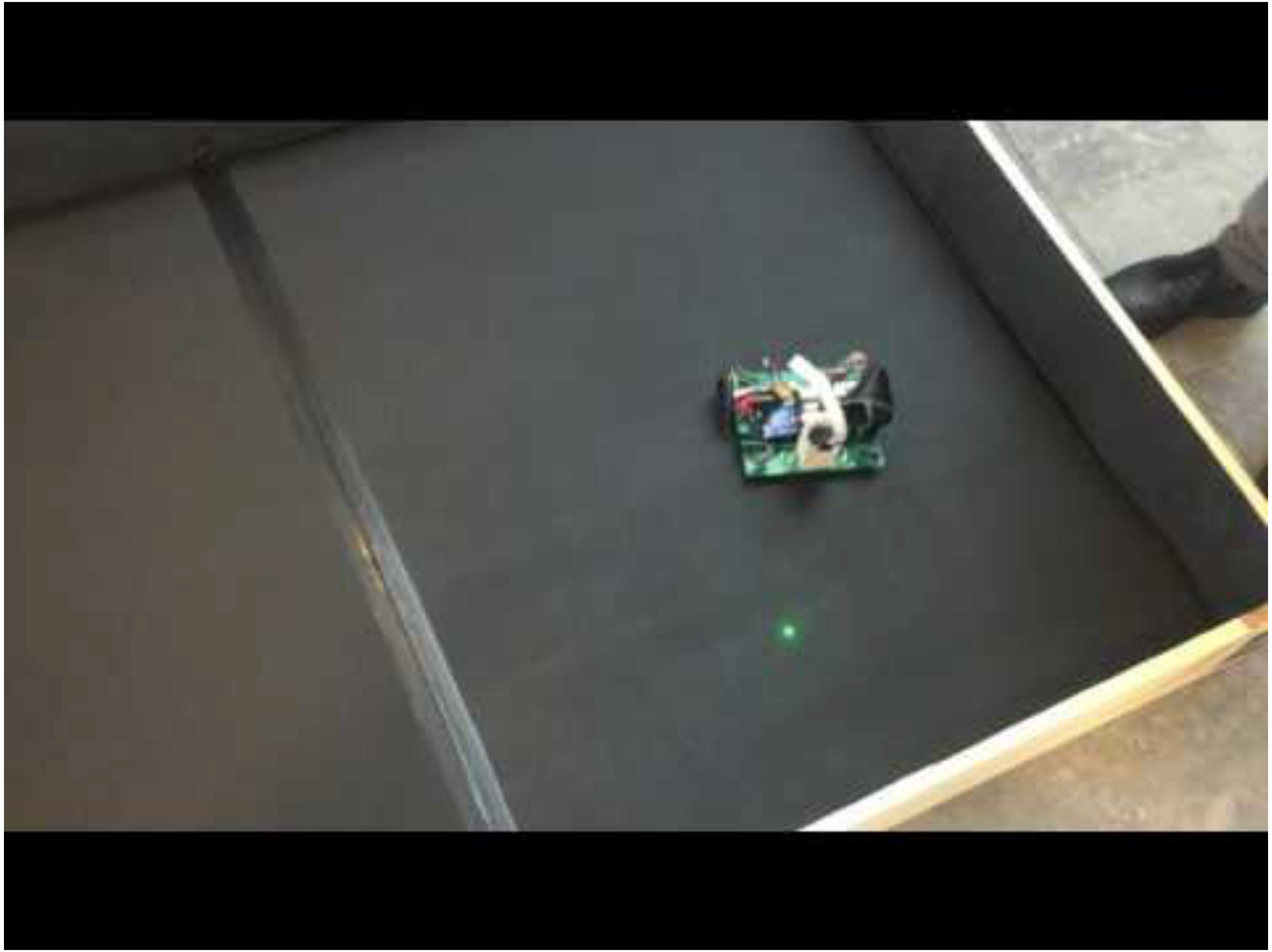
https://chess.eecs.berkeley.edu/eecs149/documentation/differentialDrive.pdf

Controls Challenges and Solutions



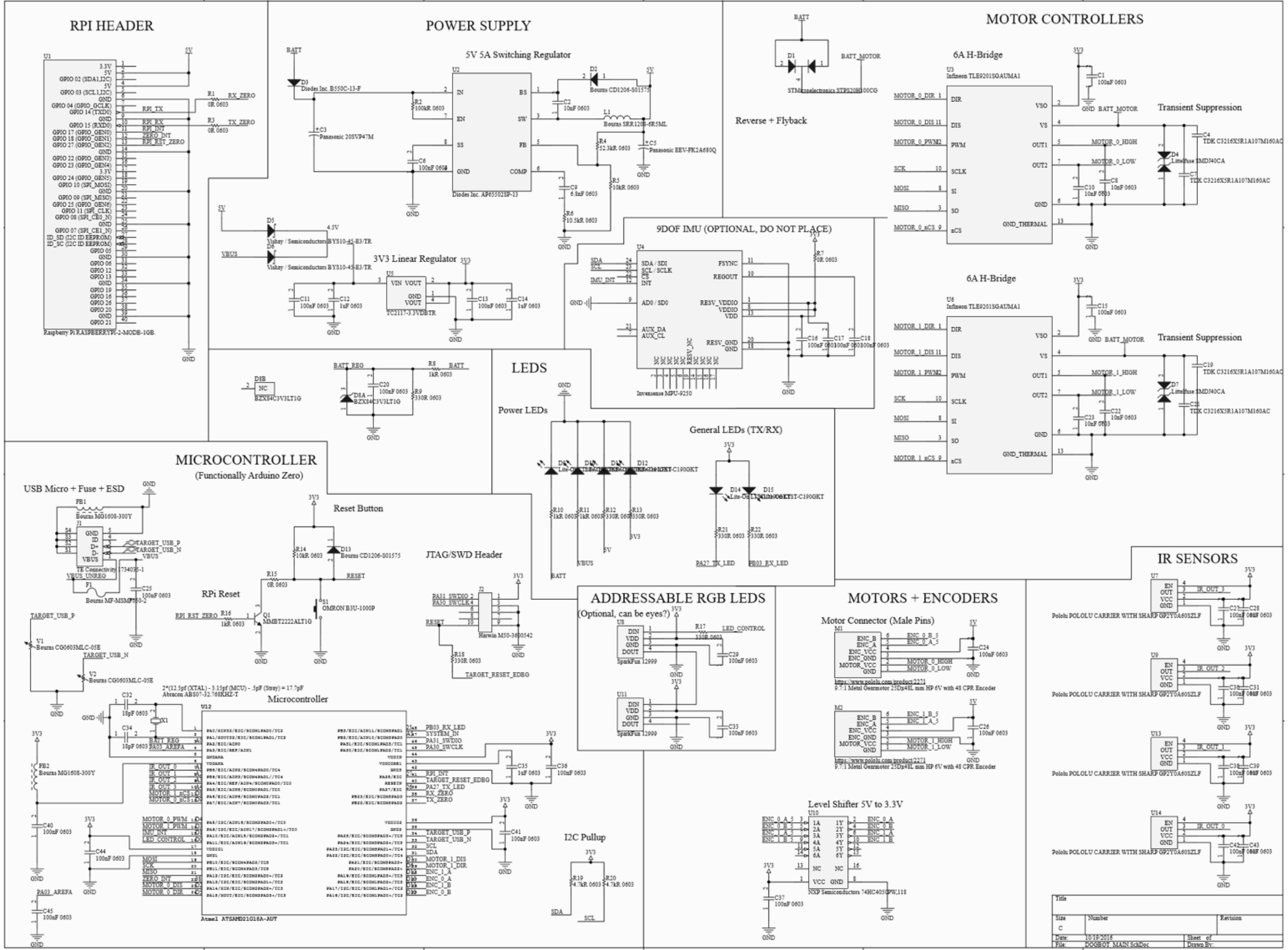
Acknowledgements

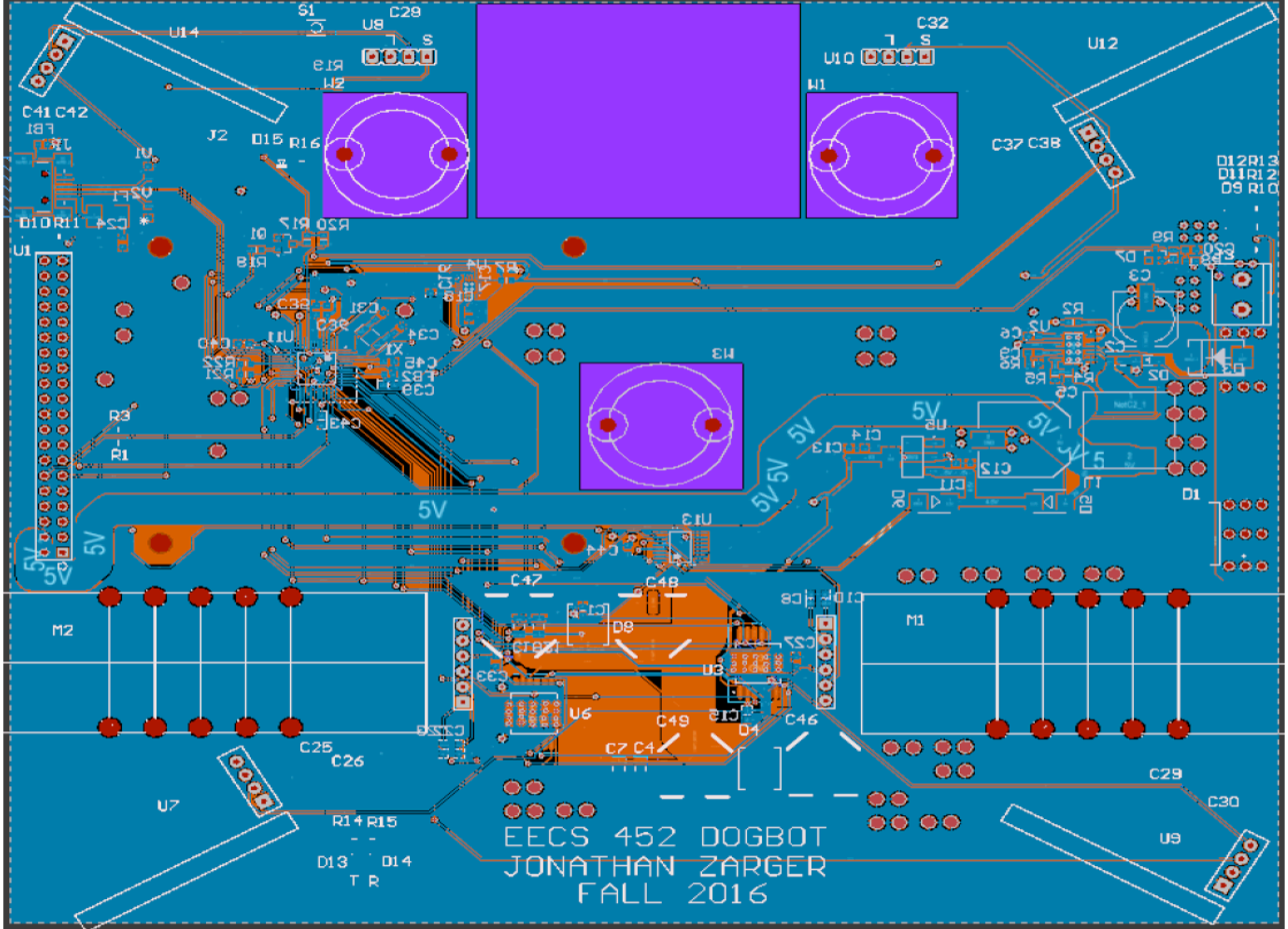
We would like to thank Professor Gregory Wakefield, Dr. Kurt Metzger, and our GSI’s Dom Calabrese and Sudheer Nuggehalli for their help and support throughout the semester



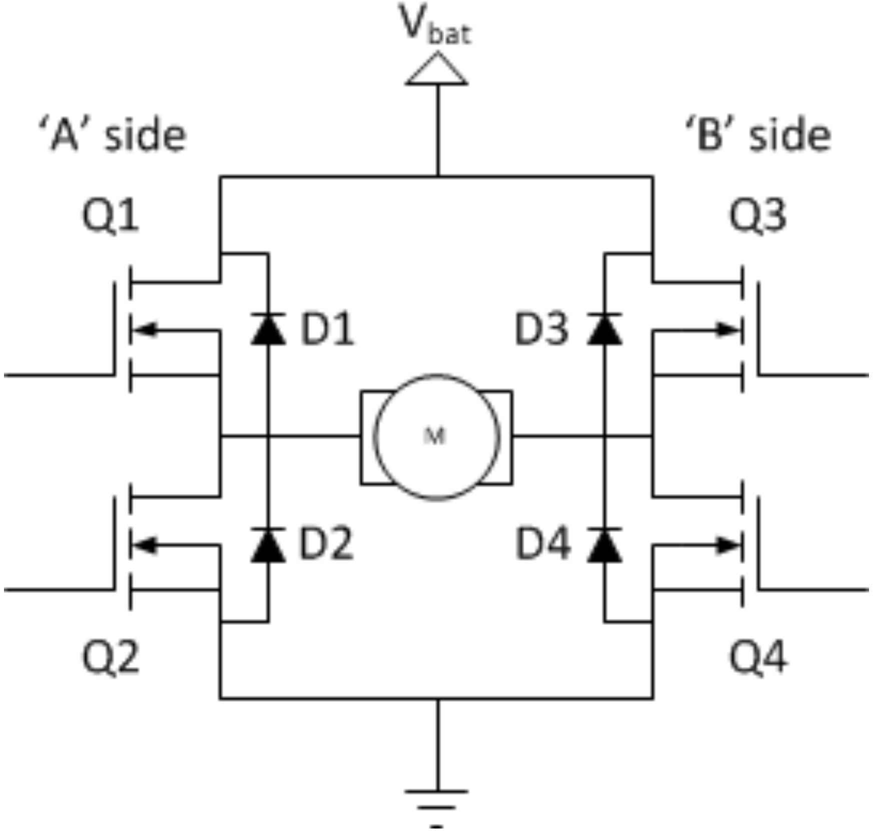
https://youtu.be/c0_0CmlvKFY

Supplementary Slides


RPI HEADER
U1
Raspberry Pi RASPBERRYPI-2-MODB-1GB.
RPI_TX
RPI_RX
RPI_INT
ZERO_INT
RPI_RST_ZERO
RX_ZERO
TX_ZERO
R1 0R 0603
R3 0R 0603
POWER SUPPLY
5V 5A Switching Regulator
BATT
D3 Diodes Inc. B550C-13-F
C3 Panasonic 20SVP47M
U2 Diodes Inc. AP65502SP-13
D2 Bourns CD1206-S01575
L1 Bourns SRR1208-6R5ML
C5 Panasonic EEV-FK2A680Q
R2 100kR 0603
R4 52.3kR 0603
R5 10kR 0603
R6 10.5kR 0603
C2 10nF 0603
C6 100nF 0603
C9 6.8nF 0603
D5 Vishay / Semiconductors BYS10-45-E3/TR
D6 Vishay / Semiconductors BYS10-45-E3/TR
4.5V
VBUS
3V3 Linear Regulator
U5 TC2117-3.3VDBTR
C11 100nF 0603
C12 1uF 0603
C13 100nF 0603
C14 1uF 0603
BATT_REG
R8 1kR 0603
R9 330R 0603
C20 100nF 0603
D8A BZX84C3V3LT1G
D8B NC BZX84C3V3LT1G
MOTOR CONTROLLERS
Reverse + Flyback
D1 STMicroelectronics STPS20H100CG
BATT_MOTOR
6A H-Bridge
U3 Infineon TLE9201SGAUMA1
U6 Infineon TLE9201SGAUMA1
MOTOR_0_DIR
MOTOR_0_DIS
MOTOR_0_PWM
MOTOR_0_nCS
MOTOR_0_HIGH
MOTOR_0_LOW
MOTOR_1_DIR
MOTOR_1_DIS
MOTOR_1_PWM
MOTOR_1_nCS
MOTOR_1_HIGH
MOTOR_1_LOW
SCK
MOSI
MISO
Transient Suppression
D4 Littelfuse SMDJ40CA
D7 Littelfuse SMDJ40CA
C4 TDK C3216X5R1A107M160AC
C7 TDK C3216X5R1A107M160AC
C19 TDK C3216X5R1A107M160AC
C21 TDK C3216X5R1A107M160AC
C1 100nF 0603
C15 100nF 0603
C8 10nF 0603
C10 10nF 0603
C22 10nF 0603
C23 10nF 0603
9DOF IMU (OPTIONAL, DO NOT PLACE)
U4 Invensense MPU-9250
R7 0R 0603
C16 100nF 0603
C17 100nF 0603
C18 100nF 0603
IMU_INT
SDA
SCL
LEDS
Power LEDs
D9 D10 D11 D12 Lite-On LTST-C190GKT
R10 1kR 0603
R11 1kR 0603
R12 330R 0603
R13 330R 0603
General LEDs (TX/RX)
D14 D15 Lite-On LTST-C190GKT
R21 330R 0603
R22 330R 0603
PA27_TX_LED
PB03_RX_LED
MICROCONTROLLER
(Functionally Arduino Zero)
USB Micro + Fuse + ESD
FB1 Bourns MG1608-300Y
J1 TE Connectivity 1734035-1
F1 Bourns MF-MSMF050-2
C25 100nF 0603
TARGET_USB_P
TARGET_USB_N
VBUS_UNREG
V1 Bourns CG0603MLC-05E
V2 Bourns CG0603MLC-05E
Reset Button
R14 10kR 0603
D13 Bourns CD1206-S01575
S1 OMRON B3U-1000P
RESET
RPi Reset
R15 0R 0603
R16 1kR 0603
Q1 MMBT2222ALT1G
JTAG/SWD Header
J2 Harwin M50-3600542
PA31_SWDIO
PA30_SWCLK
R18 330R 0603
TARGET_RESET_EDBG
2*(12.5pf (XTAL) - 3.15pf (MCU) - .5pF (Stray) = 17.7pF
Abracon ABS07-32.768KHZ-T
C32 18pF 0603
C34 18pF 0603
X1
Microcontroller
U12 Atmel ATSAMD21G18A-AUT
FB2 Bourns MG1608-300Y
C35 1uF 0603
C36 100nF 0603
C40 100nF 0603
C41 100nF 0603
C44 100nF 0603
C45 100nF 0603
PA03_AREFA
I2C Pullup
R19 4.7kR 0603
R20 4.7kR 0603
ADDRESSABLE RGB LEDS
(Optional, can be eyes?)
U8 SparkFun 12999
U11 SparkFun 12999
R17 330R 0603
LED_CONTROL
C29 100nF 0603
C33 100nF 0603
MOTORS + ENCODERS
Motor Connector (Male Pins)
M1
M2
https://www.pololu.com/product/2271
9.7:1 Metal Gearmotor 25Dx48L mm HP 6V with 48 CPR Encoder
C24 100nF 0603
C26 100nF 0603
ENC_0_B_5
ENC_0_A_5
ENC_1_B_5
ENC_1_A_5
Level Shifter 5V to 3.3V
U10 NXP Semiconductors 74HC4050PW,118
C37 100nF 0603
ENC_0_A
ENC_0_B
ENC_1_A
ENC_1_B
IR SENSORS
U7 U9 U13 U14 Pololu POLOLU CARRIER WITH SHARP GP2Y0A60SZLF
IR_OUT_3
IR_OUT_2
IR_OUT_1
IR_OUT_0
C27 C28 C30 C31 C38 C39 C42 C43 100nF 0603
GND
3V3
5V
Title
Size
C
Number
Revision
Date: 10/19/2016
Sheet of
File: DOGBOT_MAIN.SchDoc
Drawn By:


EECS 452 DOGBOT
JONATHAN ZARGER
FALL 2016
U14
S1
U8
C28
U10
C32
U12
W2
W1
W3
C41 C42
J2
D15 R16
C37 C38
D12 R13
D11 R12
D9 R10
D10 R11
C24
U1
R20 R17
R18
U11
C34
X1
C45
C39
C43
R3
R1
R9
C3
R2
U2
C6
R5
C5
D2
D3
5V
U5
C14
C13
C12
C11
D6
D5
D1
U13
C44
C47
C48
D8
C27
U3
U6
C49
C46
C7 C4
D4
M2
M1
C25
C26
U7
R14 R15
D13
D14
T R
C29
C30
U9

H-Bridge



http://modularcircuits.com/blog/wp-content/
uploads/2011/10/image7.png

Controls Implementation Details

Naive implementation of state observer

Loop time of about 5ms (200Hz)

Image Processing Used Before Milestone 2

Working Image Processing Code on Raspberry Pi

- RGB -> HSV -> HSV (Green Mask)
- HSV -> GrayScale -> Blur -> Find Contours -> Find Moment (average/center)
- Display Center on masked HSV image
  - Works on pure black background
  - Works with moving laser (line instead of point in image)