Navigation and Thrust Systems for AUVSI RoboBoat

Team: Michael S. Barnes, Evan J. Dinelli, Daniel R. Van de Water
Advisors: Dr. Gary Dempsey and Mr. Nick Schmidt

BRADLEY University

Department of Electrical and Computer Engineering

April 26th, 2016

Presentation Outline

- Background
- Evan Dinelli
    - Navigation Subsystem
    - Remote Control (RC) Unit
- Dan Van de Water
    - Motor Control Unit
- Michael Barnes
    - Power Transistors
- Summary and Conclusions
- Questions and Answers (Q & A)

Presentation Outline

- Background
  - History
  - Objective
  - Block Diagram
  - Division of Labor
- Evan Dinelli
- Dan Van de Water
- Michael Barnes
- Summary and Conclusions
- Questions and Answers (Q & A)

History

- AUVSI – Association for Unmanned Vehicle System International
  - International RoboBoat Competition
  - Bradley University has attended twice

AUVSIfoundation
ASSOCIATION FOR UNMANNED
VEHICLE SYSTEMS INTERNATIONAL

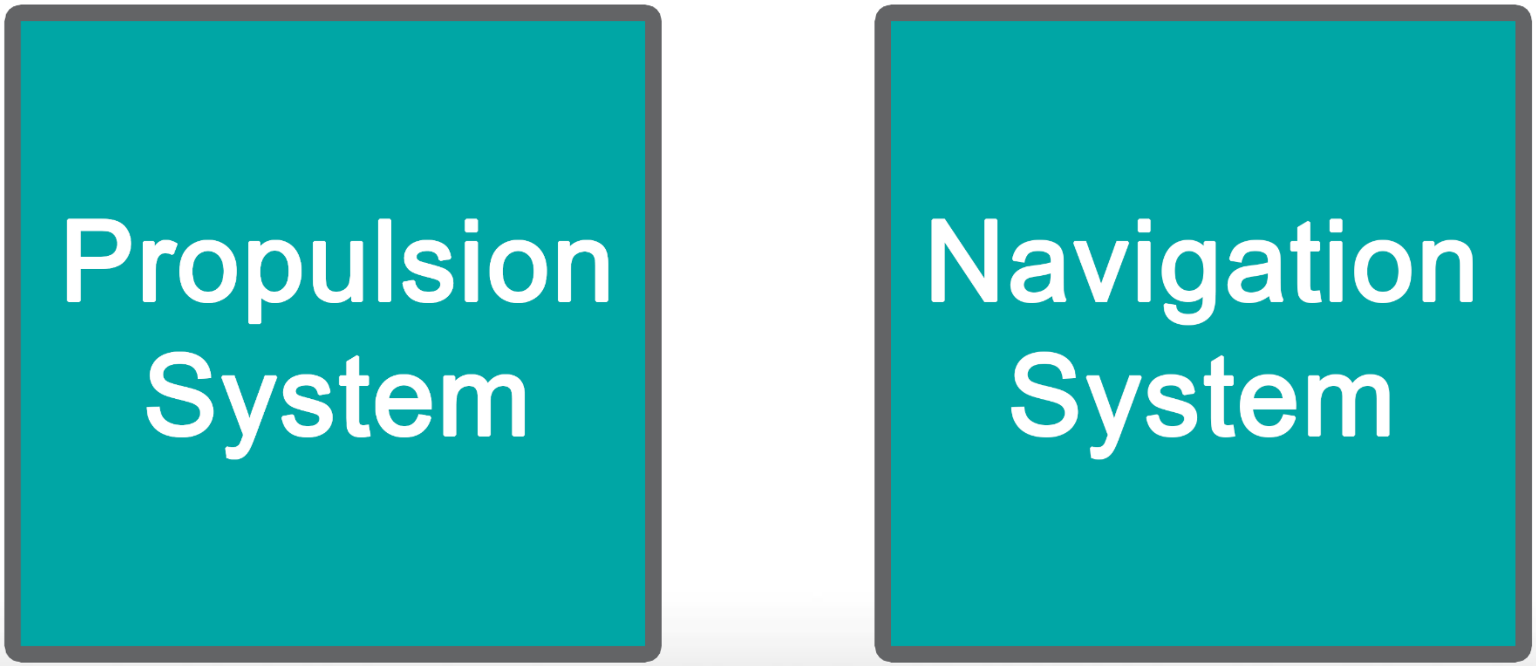
Images taken from [1] and [2].

Objective

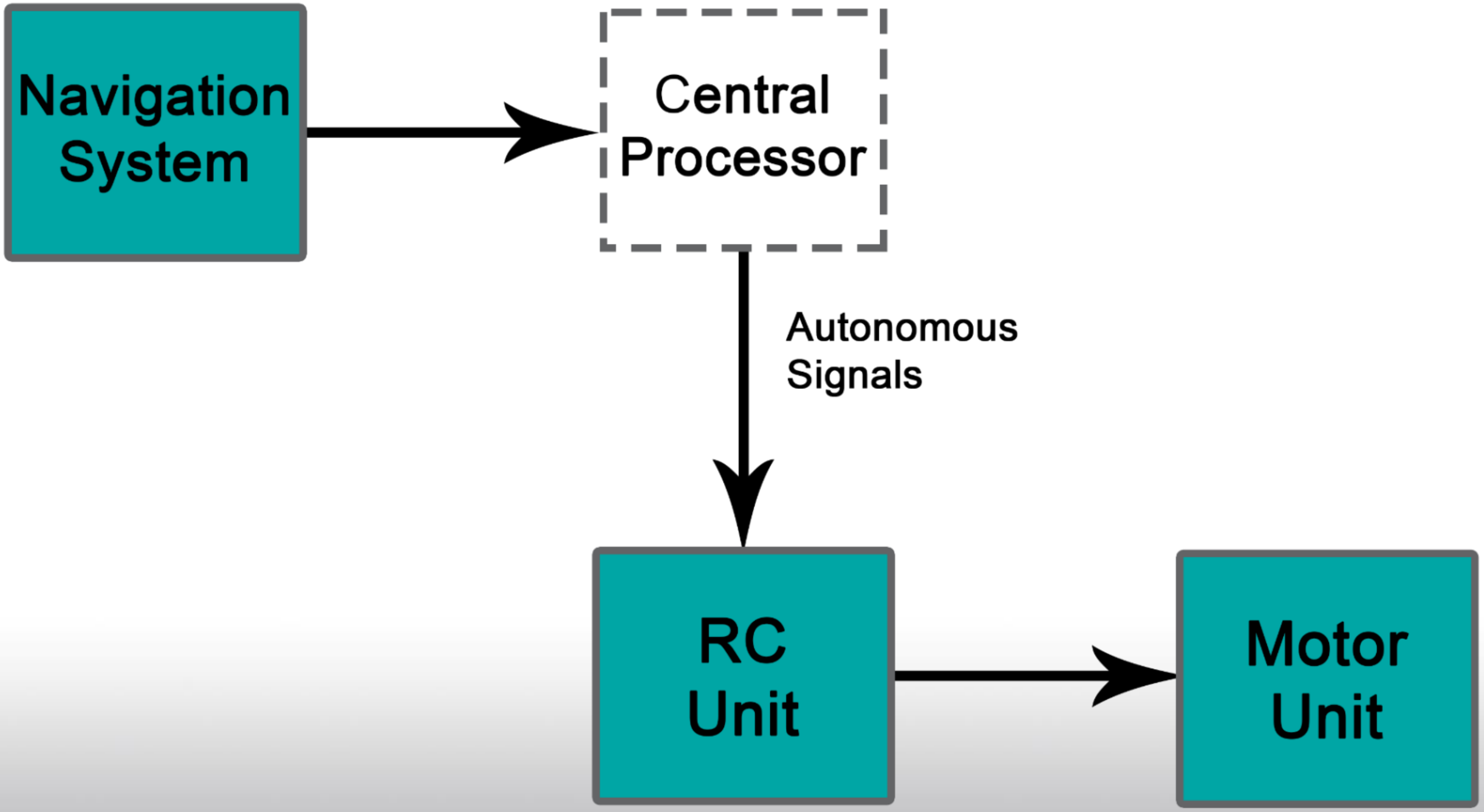
- Design and Build a System that Serves as the Framework for RoboBoat

Images taken from [1] and [2].

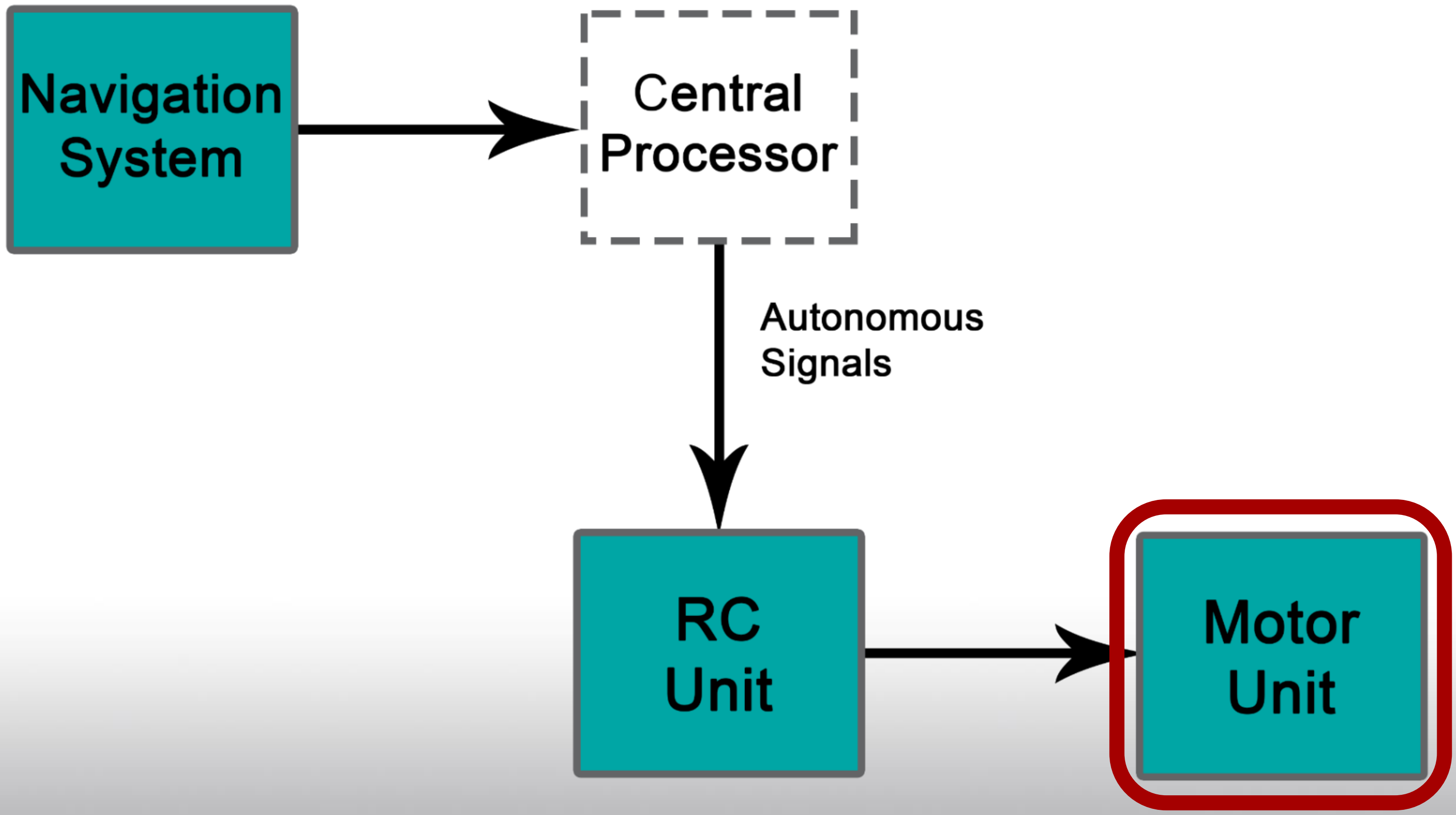
Block Diagram



Block Diagram

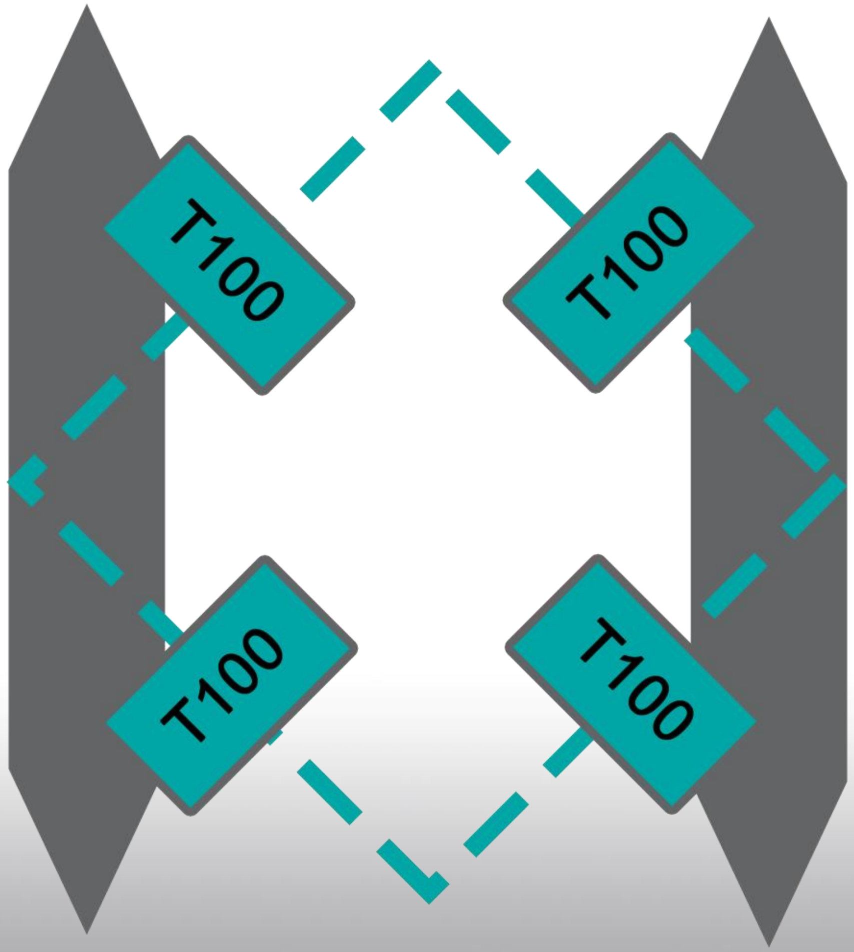


Block Diagram



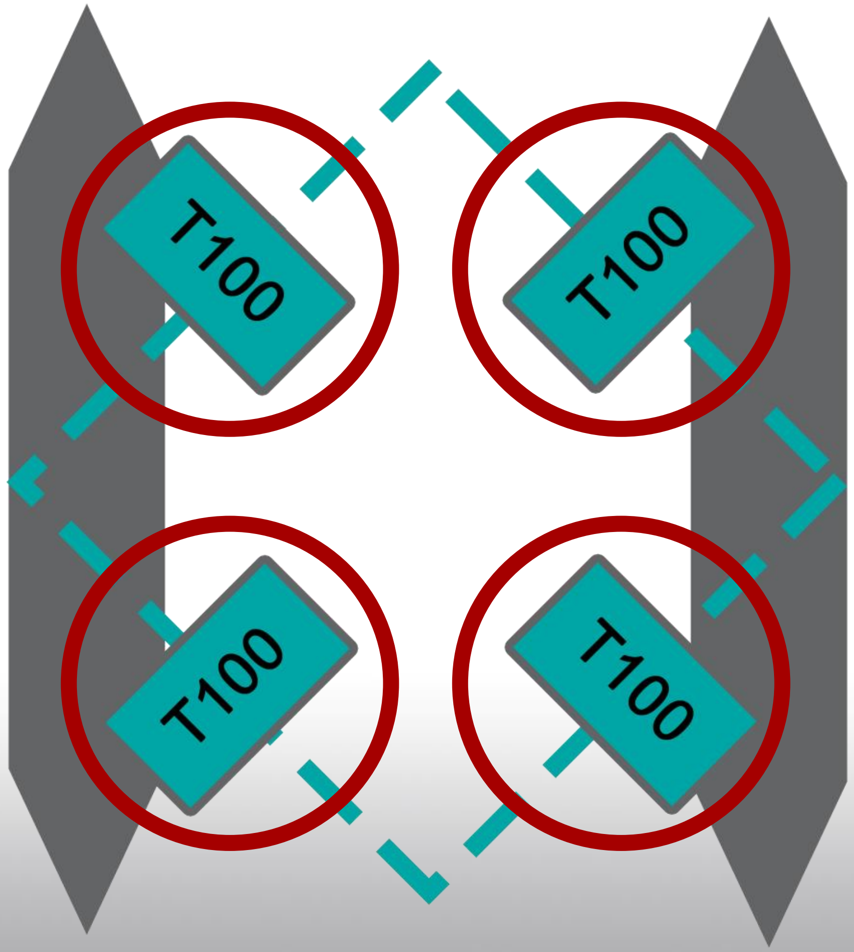
Boat

- Catamaran
  - Pontoon design
- Motor locations
- T100 Thrusters



Boat

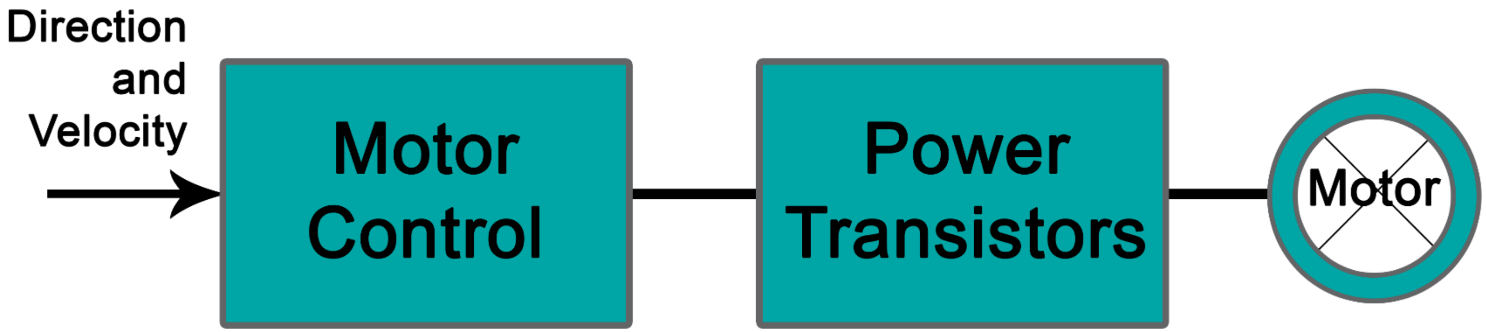
- Catamaran
  - Pontoon design
- Motor locations
- T100 Thrusters



T100 Thrusters

Image taken from [3].

Motor Control Unit



Division of Labor

Task	Person Assigned to Task
Navigation System	Evan
Remote Control	Evan
Motor Control	Daniel
Power Transistors	Michael

Functional Requirements

Functional Requirements	Specifications
GPS and Compass	GPS Accuracy: < ±2 m
	Compass Accuracy: < ±2°
Remote Control	Rate: ≥ 5 Commands / sec
	Mode Signal
	Software Kill Signal
Motor Control	Rate: ≥ 5 Commands / sec
	Physical Kill Switch

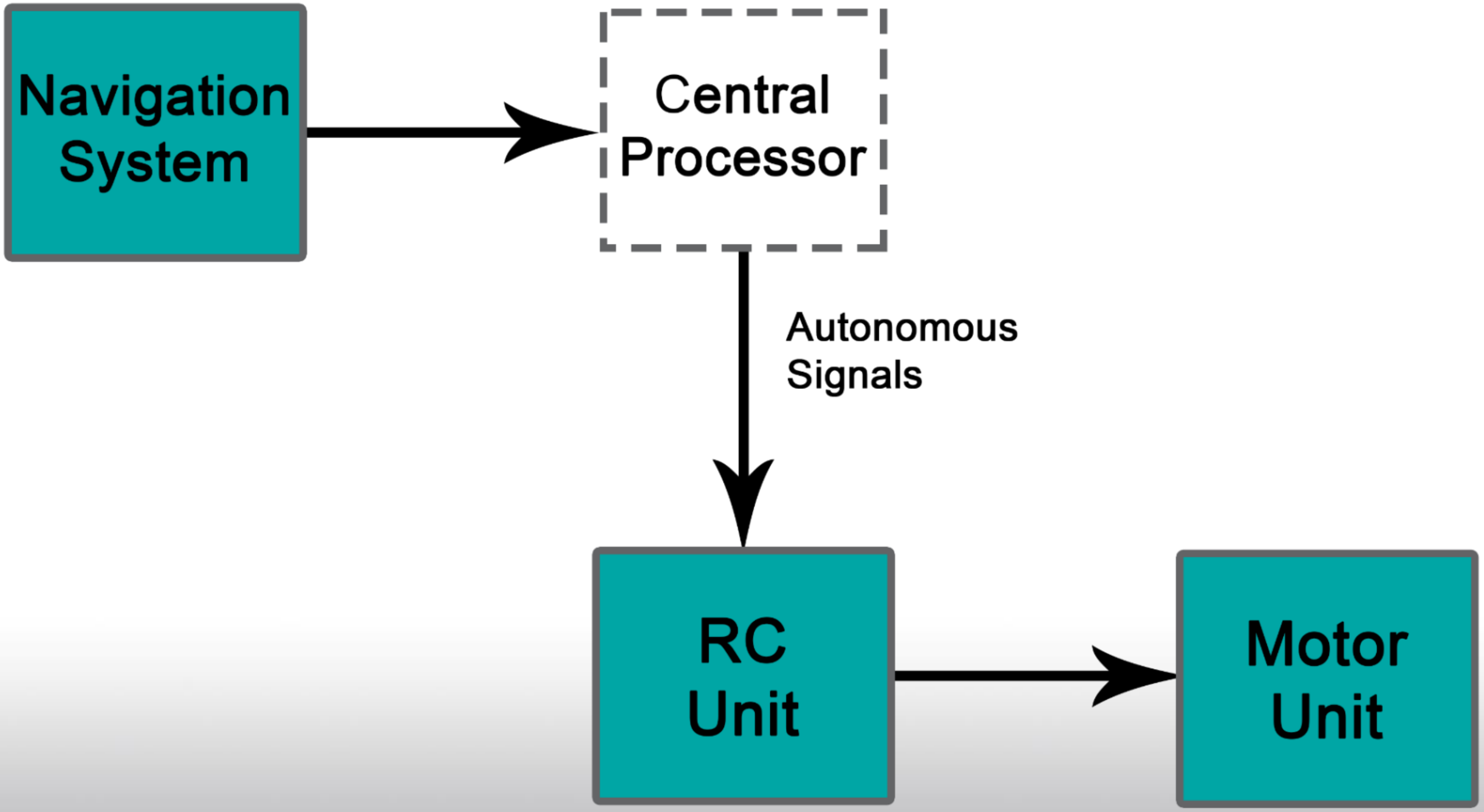
Presentation Outline

- Background
- Evan Dinelli
  - Navigation Subsystem
    - Specifications
    - Block Diagram
    - GPS
    - Compass
  - RC Unit
- Dan Van de Water
- Michael Barnes
- Summary and Conclusions
- Questions and Answers (Q & A)

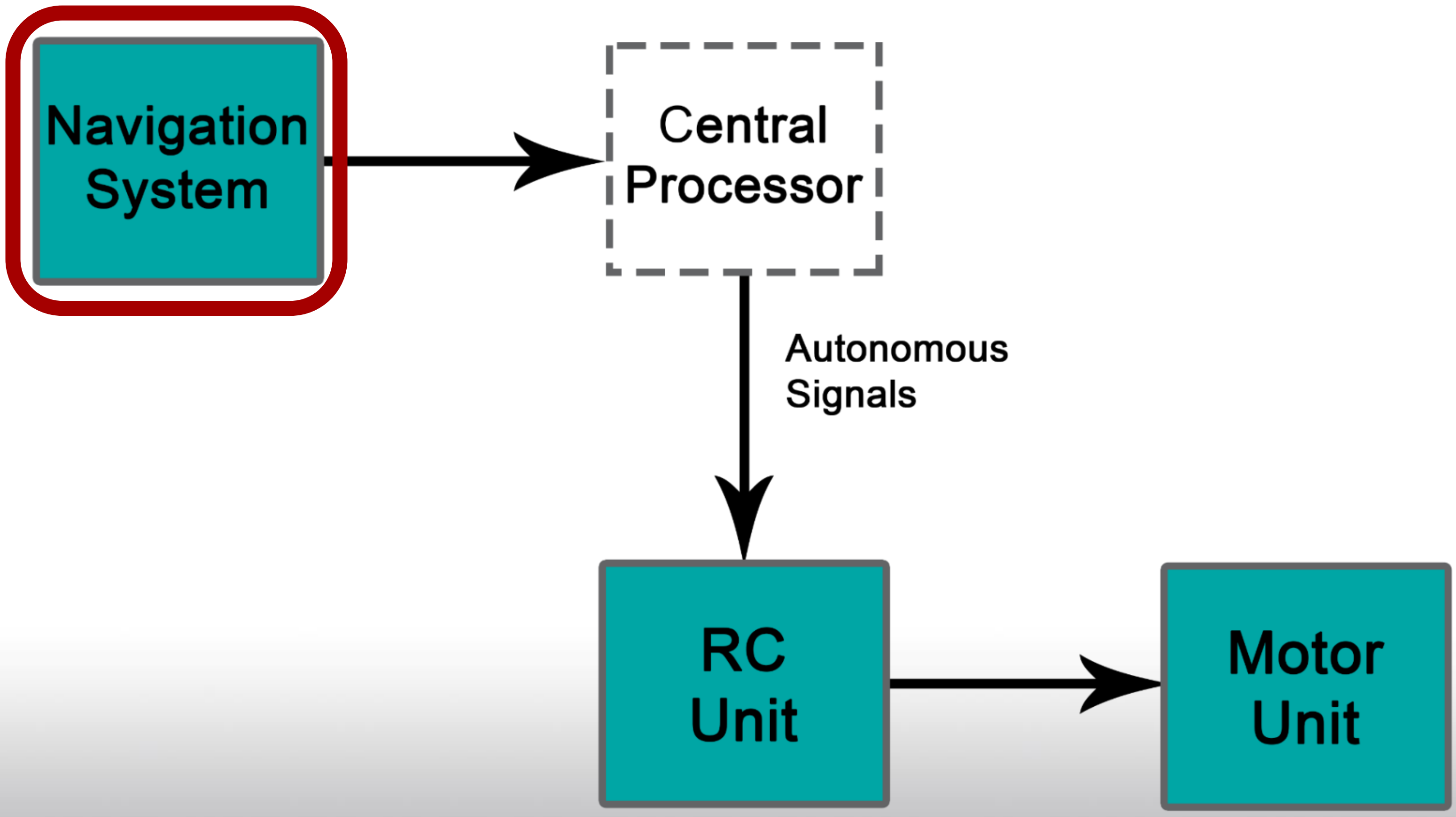
Functional Requirements

Functional Requirements	Specifications
GPS and Compass	GPS Accuracy: < ±2 m
	Compass Accuracy: < ±2°
Remote Control	Rate: ≥ 5 Commands / sec
	Mode Signal
	Software Kill Signal
Motor Control	Rate: ≥ 5 Commands / sec
	Physical Kill Switch

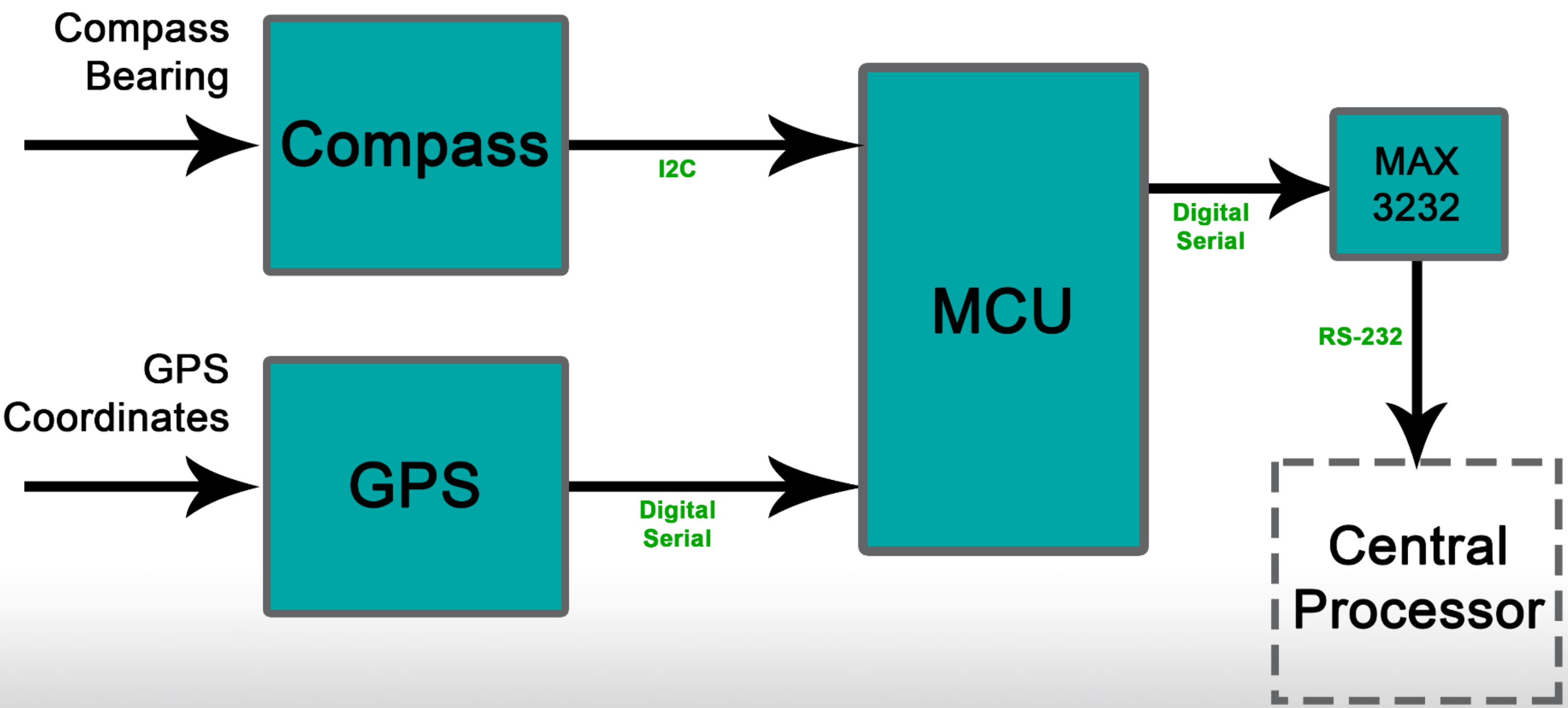
Block Diagram



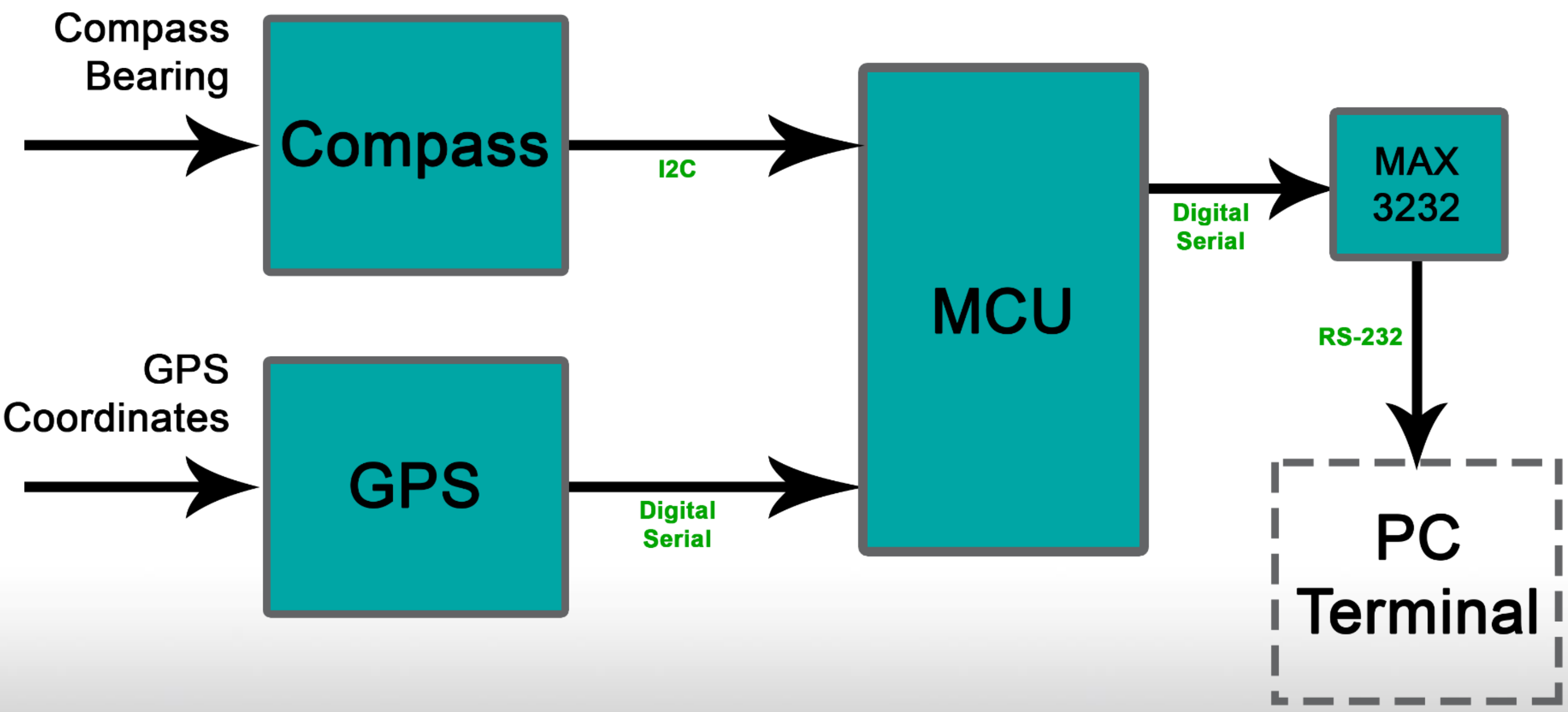
Block Diagram



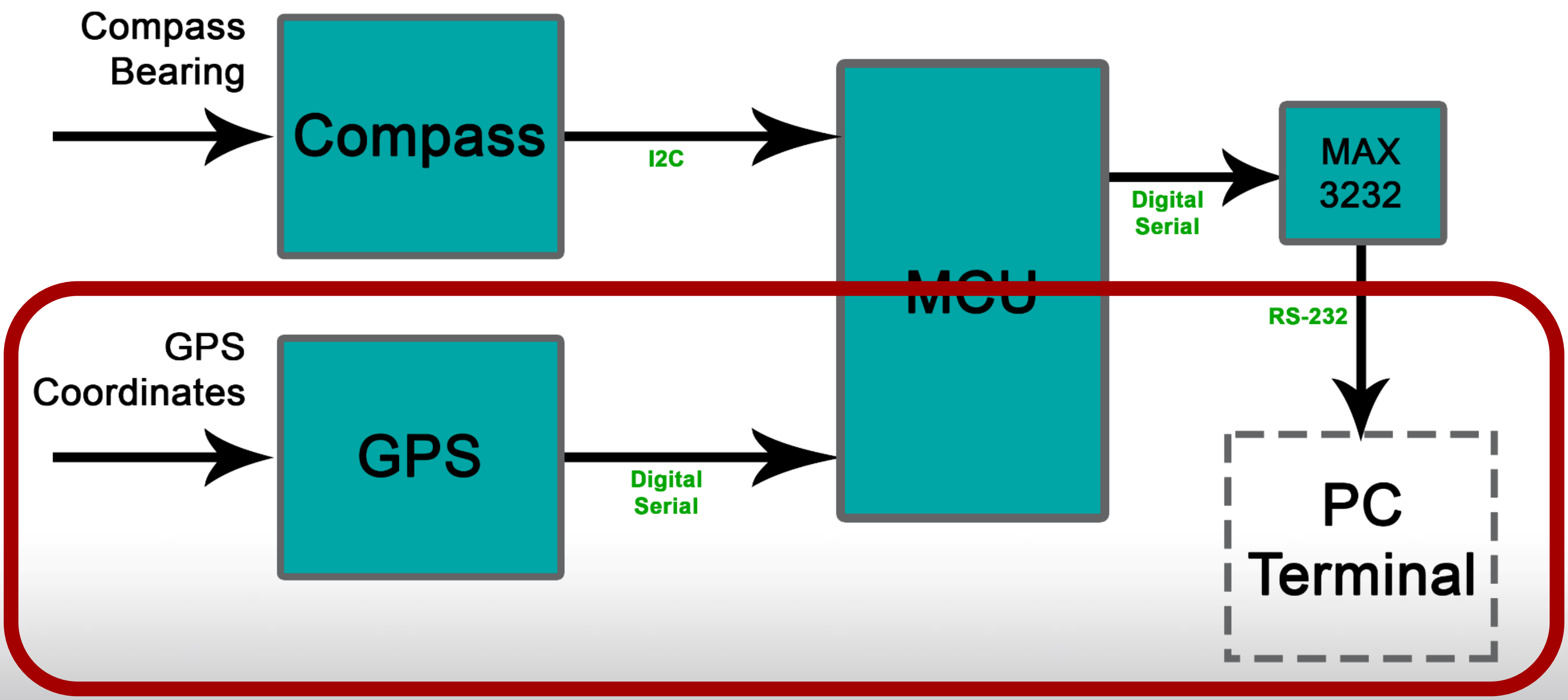
Block Diagram: Navigation



Block Diagram: Navigation



Block Diagram: Navigation



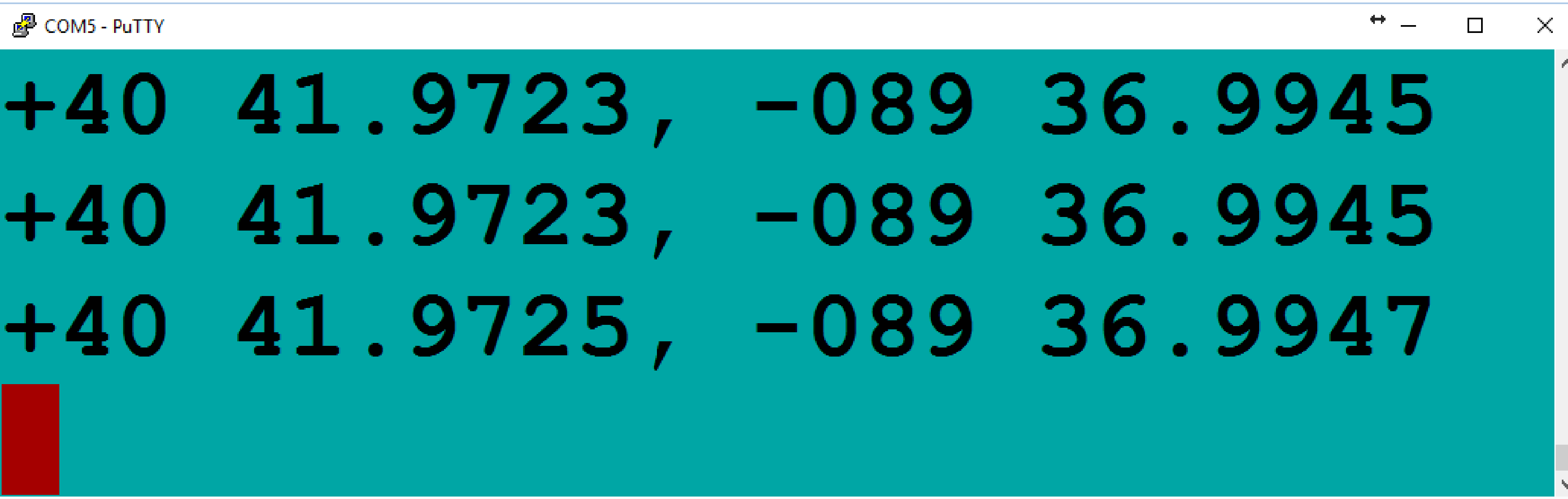
Raw GPS Data

COM5 - PuTTY

```
$GPVTG,350.24,T,,M,1.03,N,1.91,K,A*36
$GPGGA,193212.000,4041.9958,N,08937.0117,W,1,8,1.25,196.5,M,-34.0,M,,*65
$GPGSA,A,3,07,27,16,26,30,08,09,23,,,,,2.28,1.25,1.91*07
$GPRMC,193212.000,A,4041.9958,N,08937.0117,W,1.06,351.05,220416,,,A*7E
$GPVTG,351.05,T,,M,1.06,N,1.97,K,A*37
$GPGGA,193213.000,4041.9959,N,08937.0117,W,1,8,1.25,196.4,M,-34.0,M,,*64
$GPGSA,A,3,07,27,16,26,30,08,09,23,,,,,2.28,1.25,1.91*07
$GPGSV,3,1,09,09,63,247,22,27,56,085,34,08,55,149,23,23,50,189,23*73
$GPGSV,3,2,09,07,44,307,44,16,35,050,43,48,25,235,30,30,17,296,46*77
$GPGSV,3,3,09,26,09,060,31*49
$GPRMC,193213.000,A,4041.9959,N,08937.0117,W,1.60,351.04,220416,,,A*7F
$GPVTG,351.04,T,,M,1.60,N,2.97,K,A*35
$GPGGA,193214.000,4041.9964,N,08937.0118,W,1,8,1.25,196.2,M,-34.0,M,,*64
$GPGSA,A,3,07,27,16,26,30,08,09,23,,,,,2.28,1.25,1.91*07
$GPRMC,193214.000,A,4041.9964,N,08937.0118,W,1.56,352.14,220416,,,A*7E
$GPVTG,352.14,T,,M,1.56,N,2.90,K,A*35
```

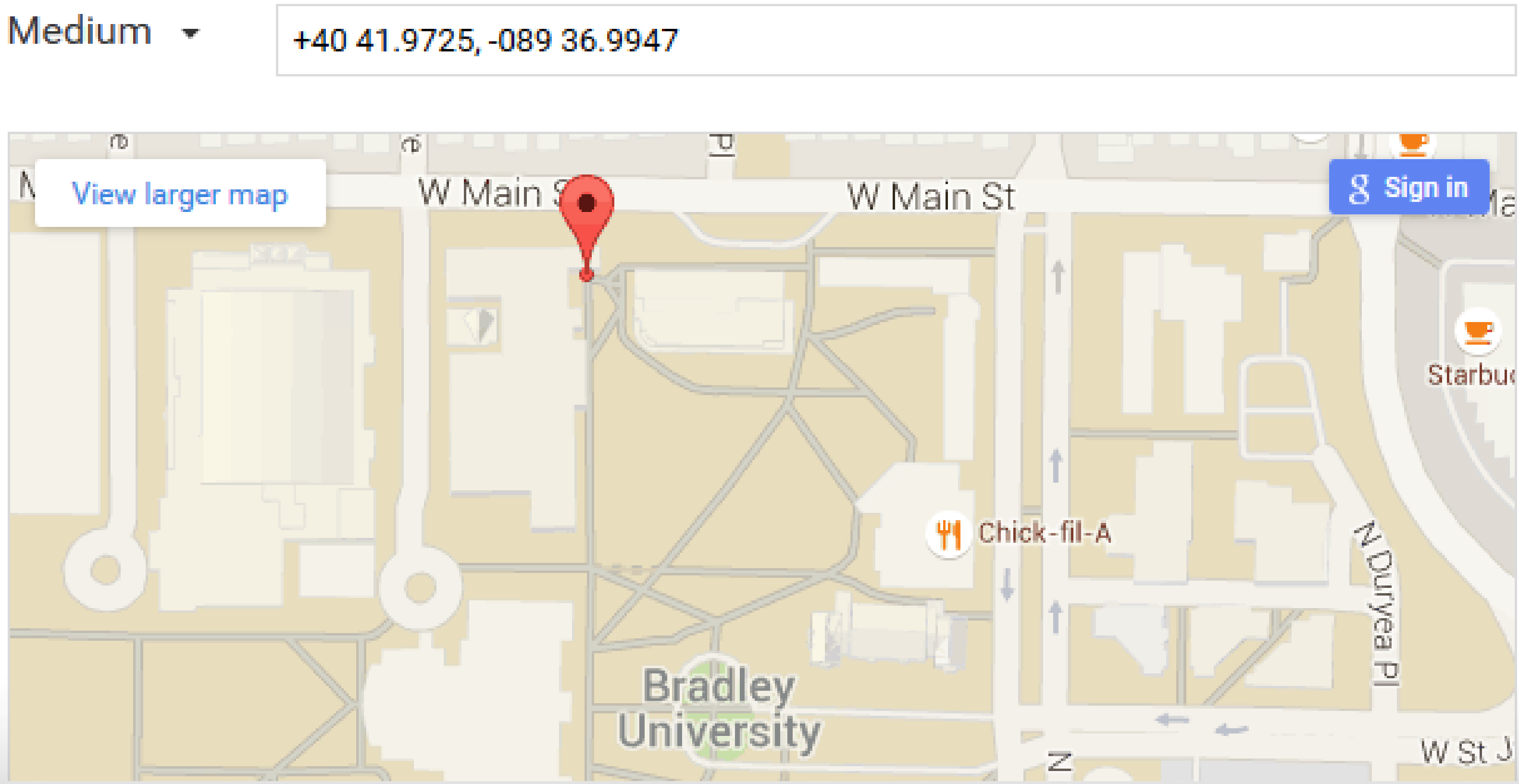
Serial Stream Directly From GPS Sensor

GPS Output



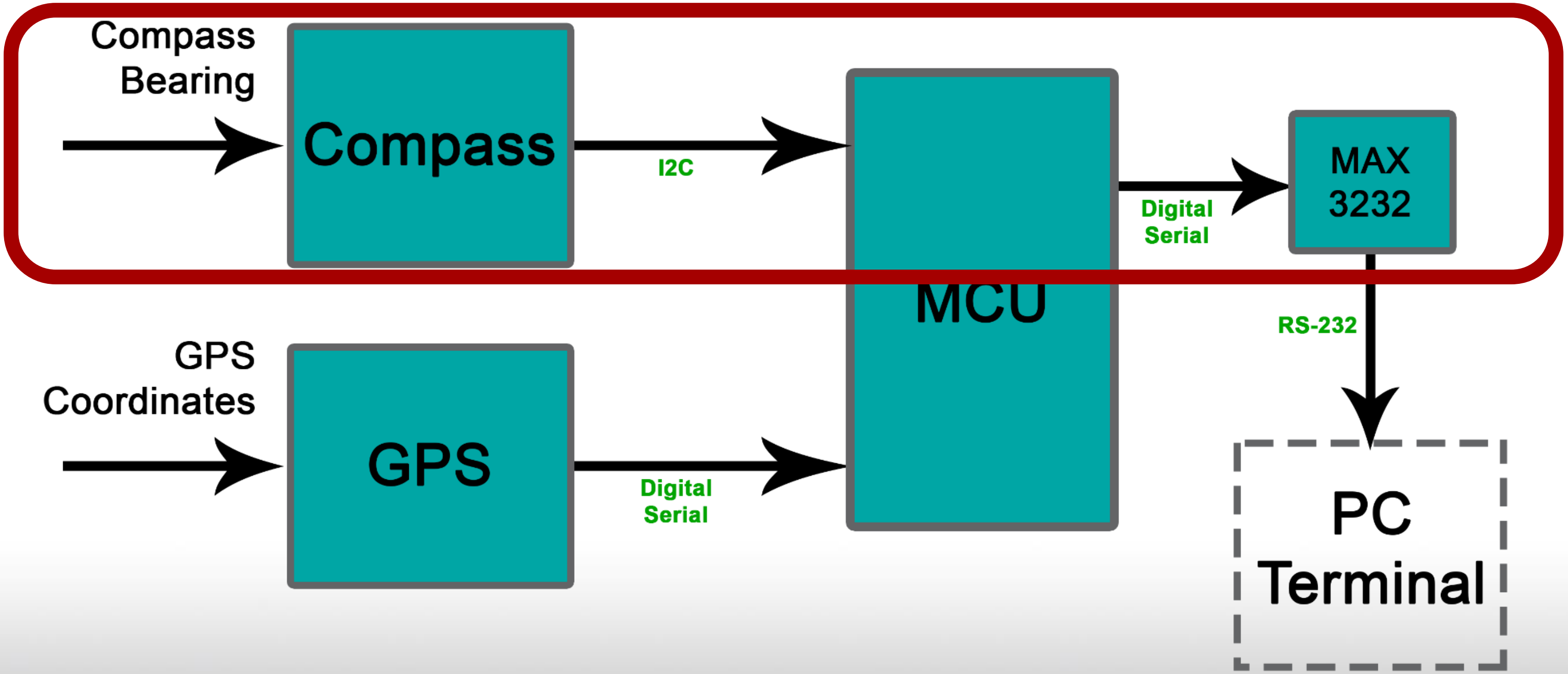
Serial Stream Directly From MCU

Verification of GPS Data

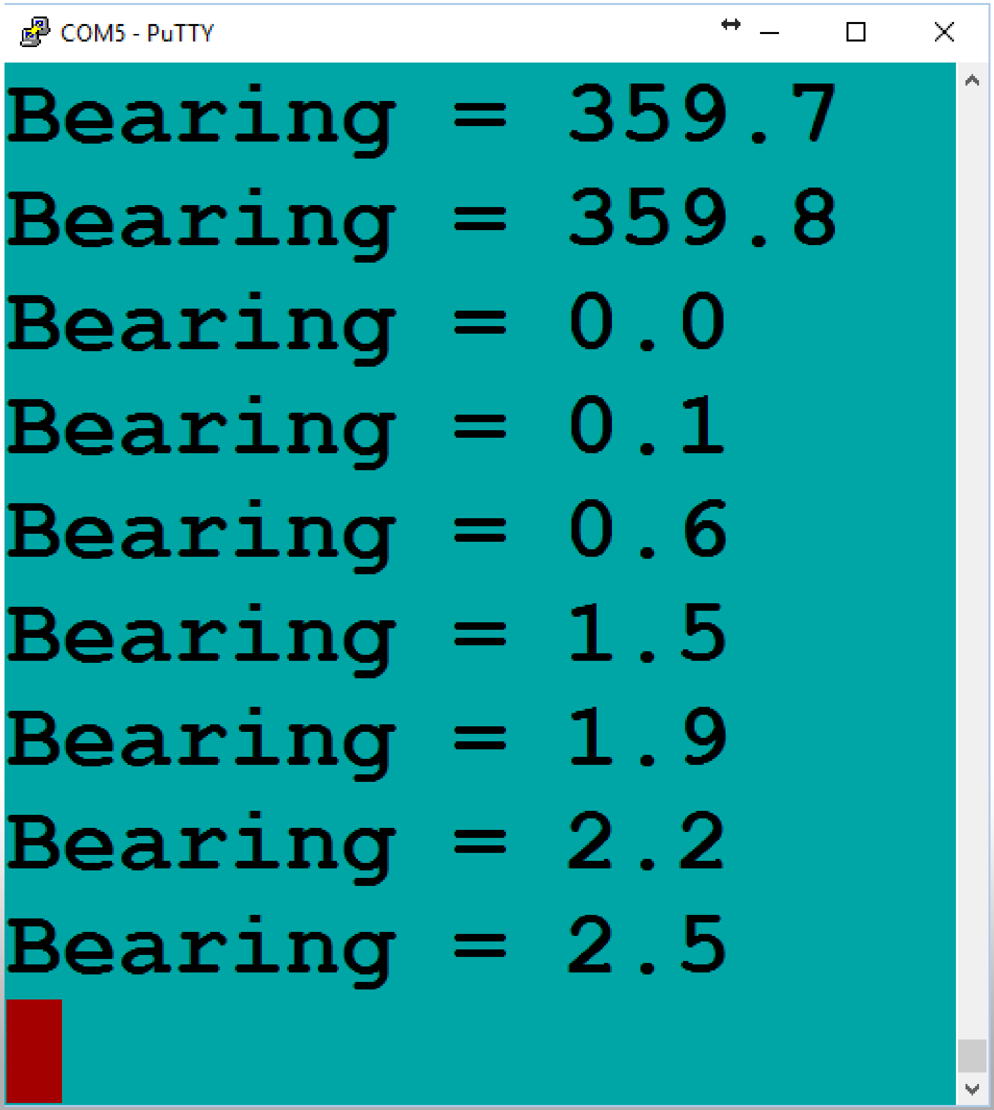


Screenshot of Google Maps

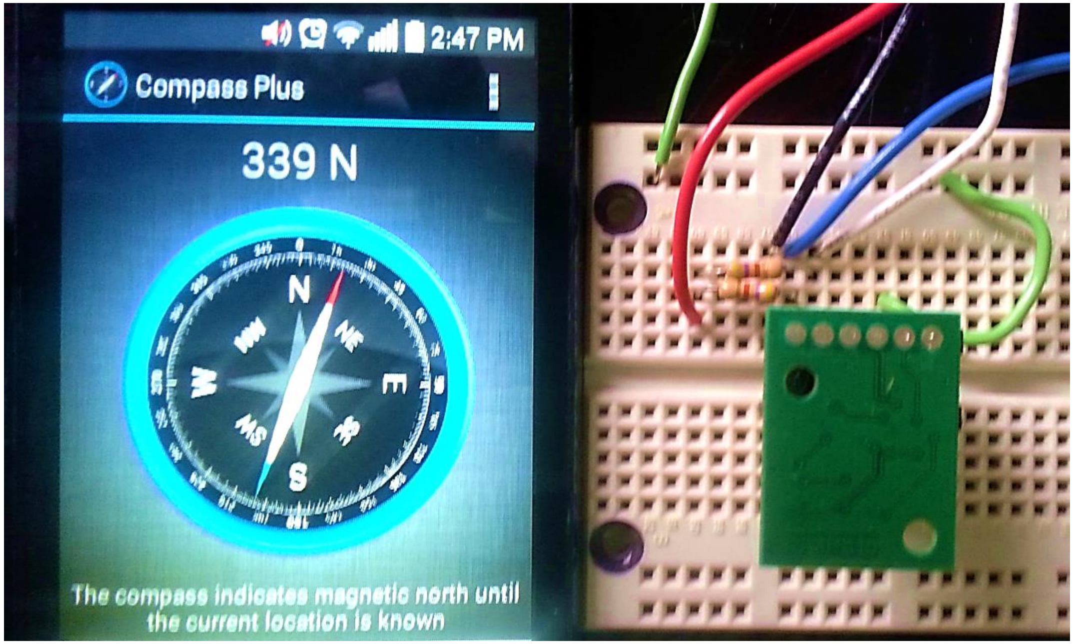
Block Diagram: Navigation

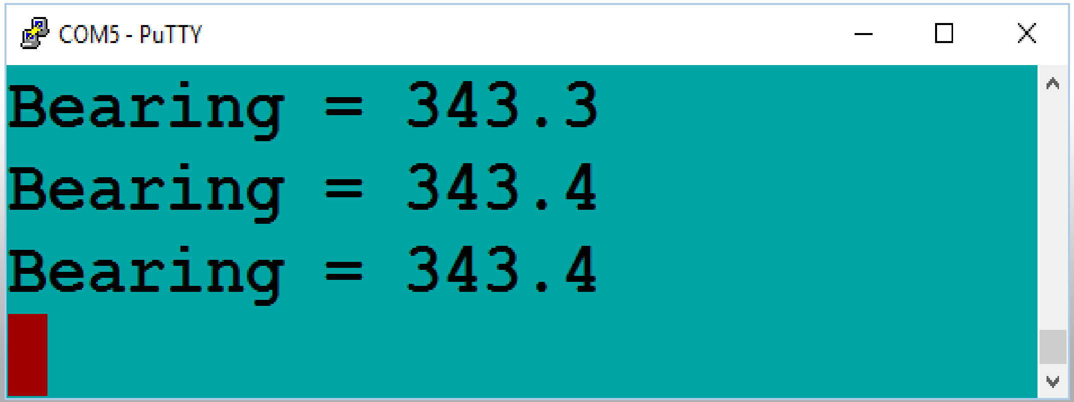


Compass Output

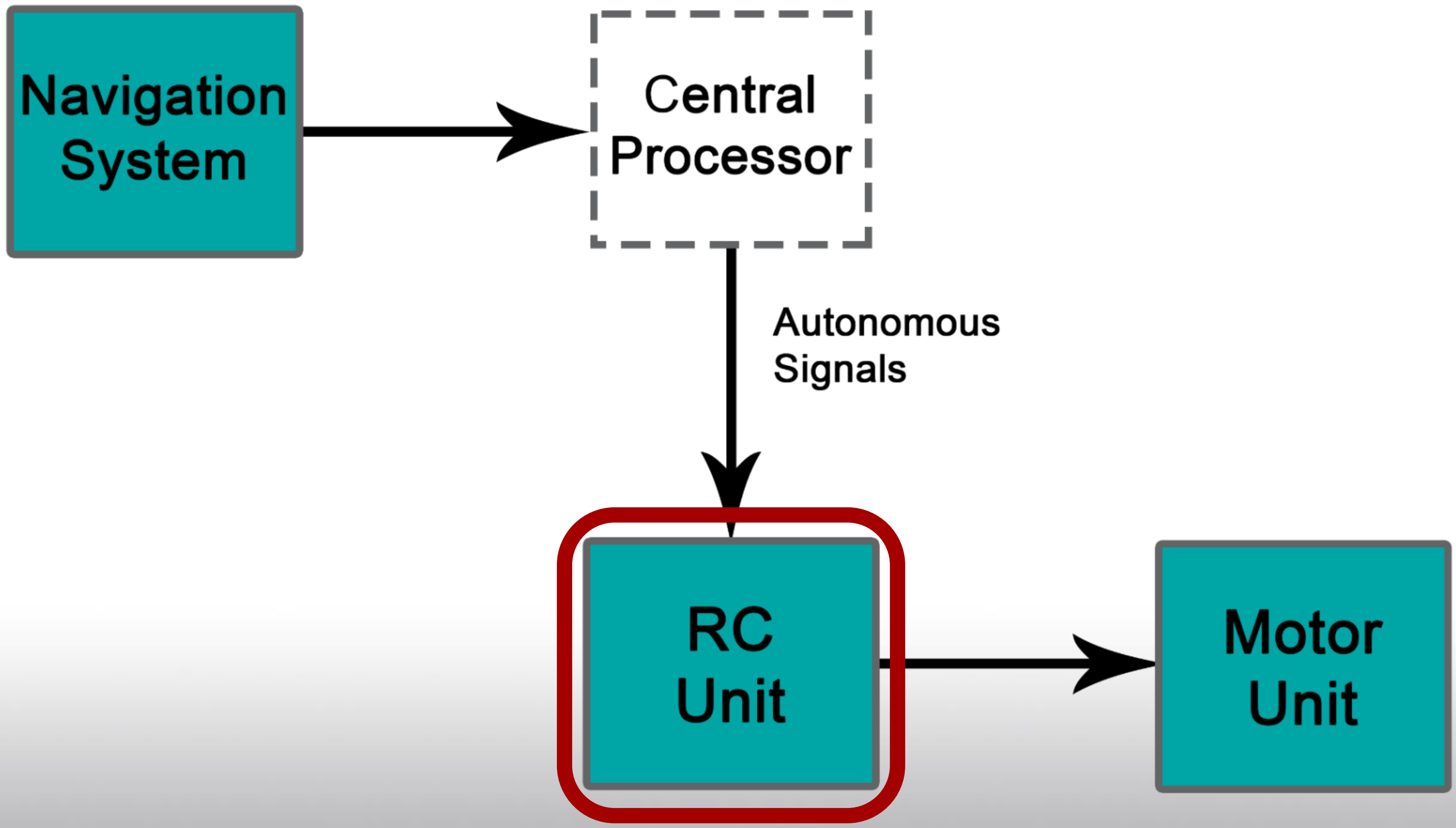


Verification of Compass Data





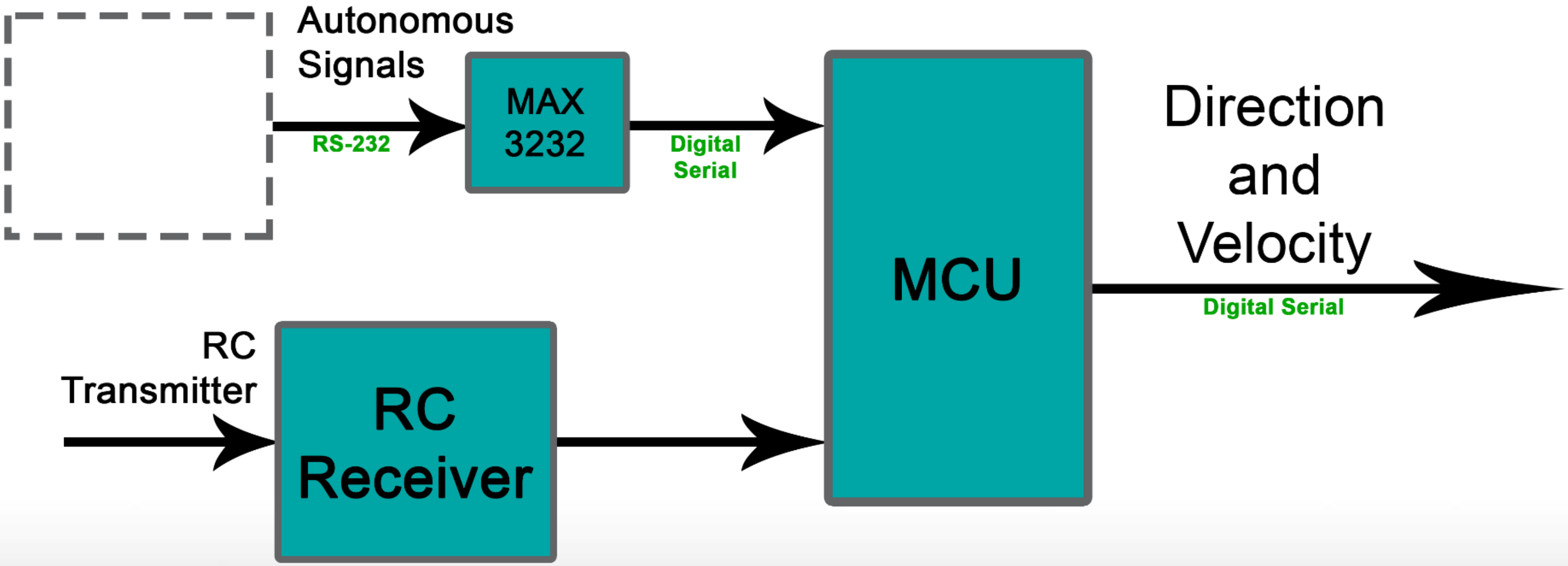
Block Diagram



Functional Requirements

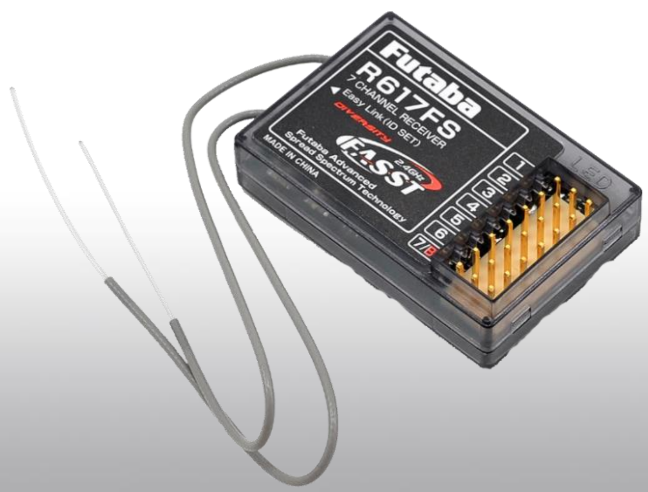
Functional Requirements	Specifications
GPS and Compass	GPS Accuracy: < ±2 m
	Compass Accuracy: < ±2°
Remote Control	Rate: ≥ 5 Commands / sec
	Mode Signal
	Software Kill Signal
Motor Control	Rate: ≥ 5 Commands / sec
	Physical Kill Switch

Block Diagram: RC Unit



RC Unit

- Futaba T6EX and the R617FS
  - 2 Joy Sticks
  - 2 Switches
  - 2.4 GHz



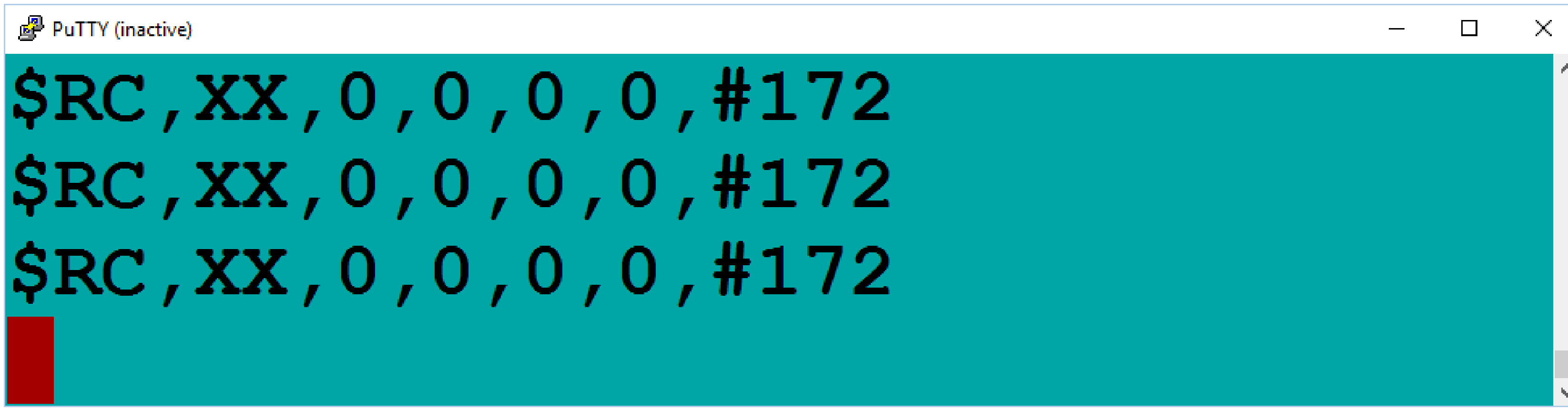
RC Unit


Futaba
DIGITAL PROPORTIONAL RADIO CONTROL SYSTEM
CH.5
A:GEAR
H:GYRO
A:TRAINER
H:IDLE-UP
AIL/ELEV/RUD
D/R
A:CH.6
FLAPS
TH.HOLD
T6EX
Computerized Airplane/Helicopter System
FASST
Mode
THR. Cut
PRESS BOTH
Select
CHANNEL
2.4GHz Spread Spectrum Technology

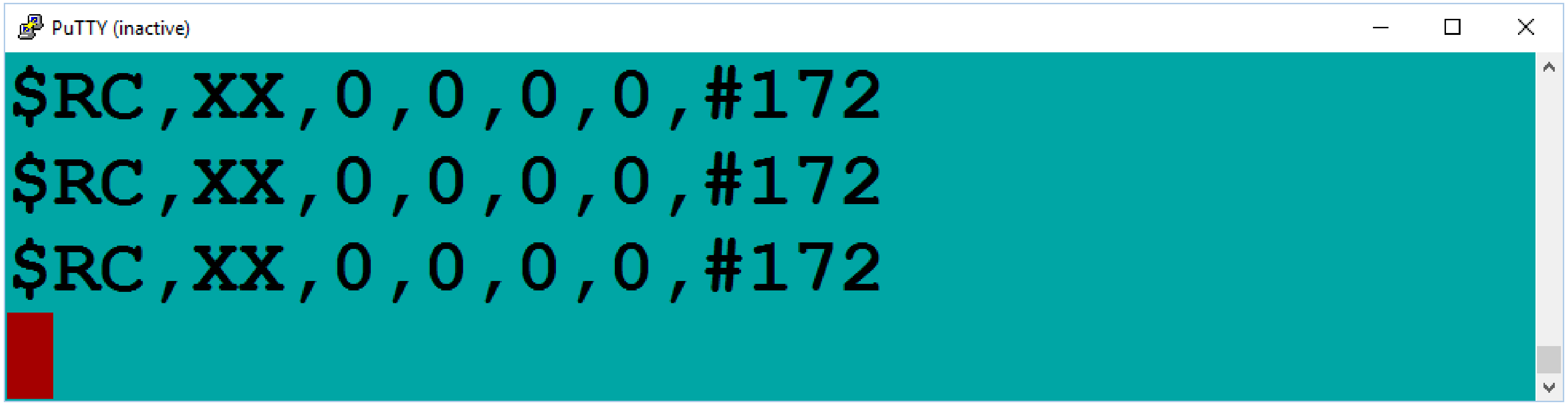
Mode Signal

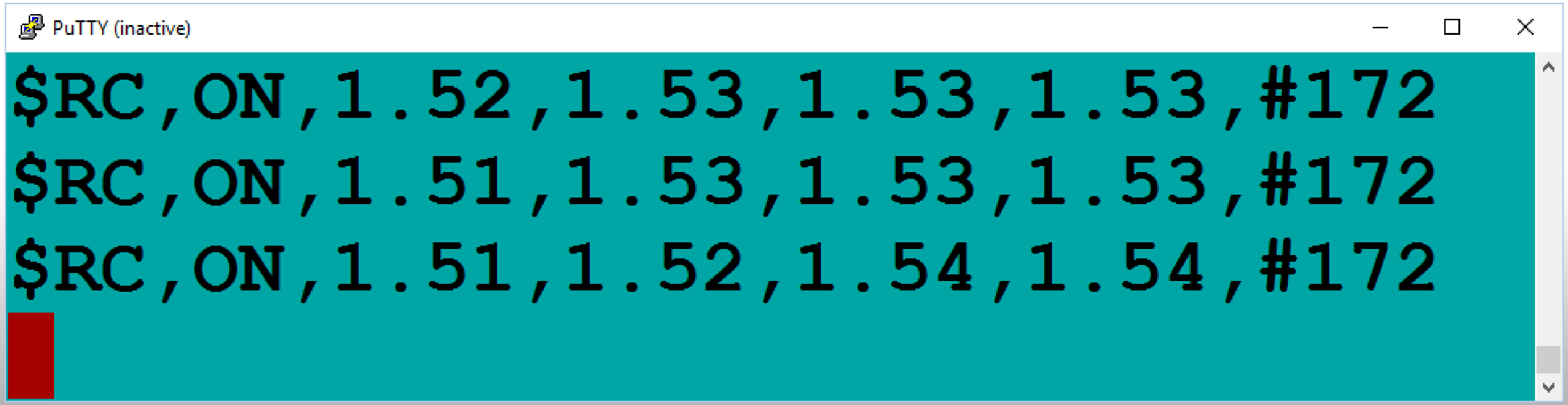

Futaba
DIGITAL PROPORTIONAL
RADIO CONTROL SYSTEM
CH.5
A:GEAR
H:GYRO
A:TRAINER
H:IDLE-UP
AIL/ELEV/
RUD
D/R
A:CH.6
FLAPS
TH.HOLD
T6EX
Computerized Airplane/Helicopter System
FASST
Mode
THR.
Cut
PRESS
BOTH
Select
CHANNEL
2.4GHz Spread Spectrum Technology

Mode Signal



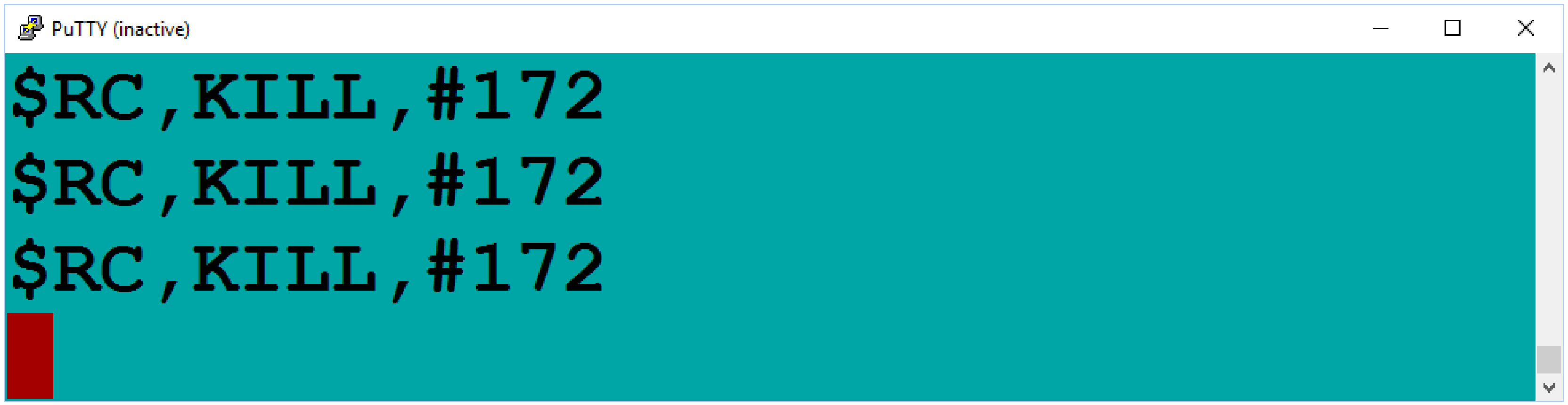
Mode Signal


PuTTY (inactive)
$RC,XX,0,0,0,0,#172
$RC,XX,0,0,0,0,#172
$RC,XX,0,0,0,0,#172

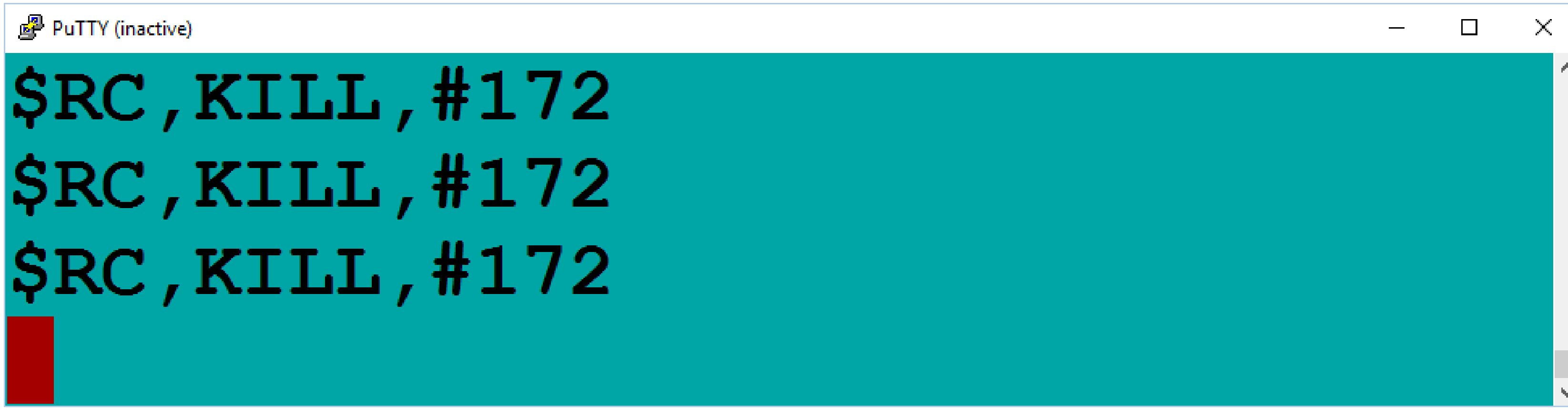

PuTTY (inactive)
$RC,ON,1.52,1.53,1.53,1.53,#172
$RC,ON,1.51,1.53,1.53,1.53,#172
$RC,ON,1.51,1.52,1.54,1.54,#172

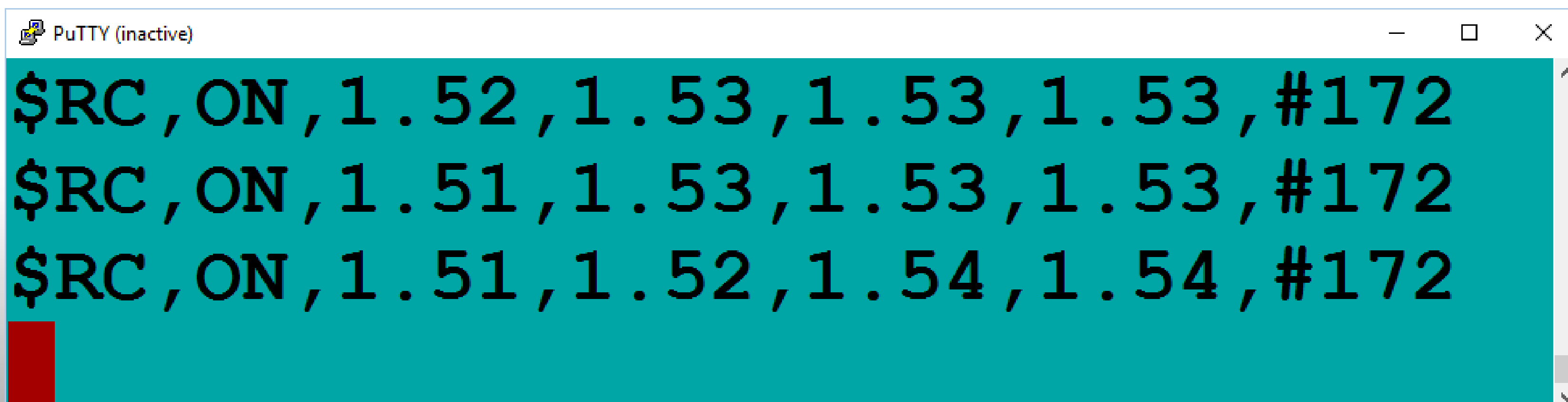
Motor Shutdown Signal

Motor Shutdown Signal



Motor Shutdown Signal





Motor Movement Commands

How the Remote Works

- Each Potentiometer Maps to a Channel, 1 - 4
- Two Switches Map to Channels 5 - 6

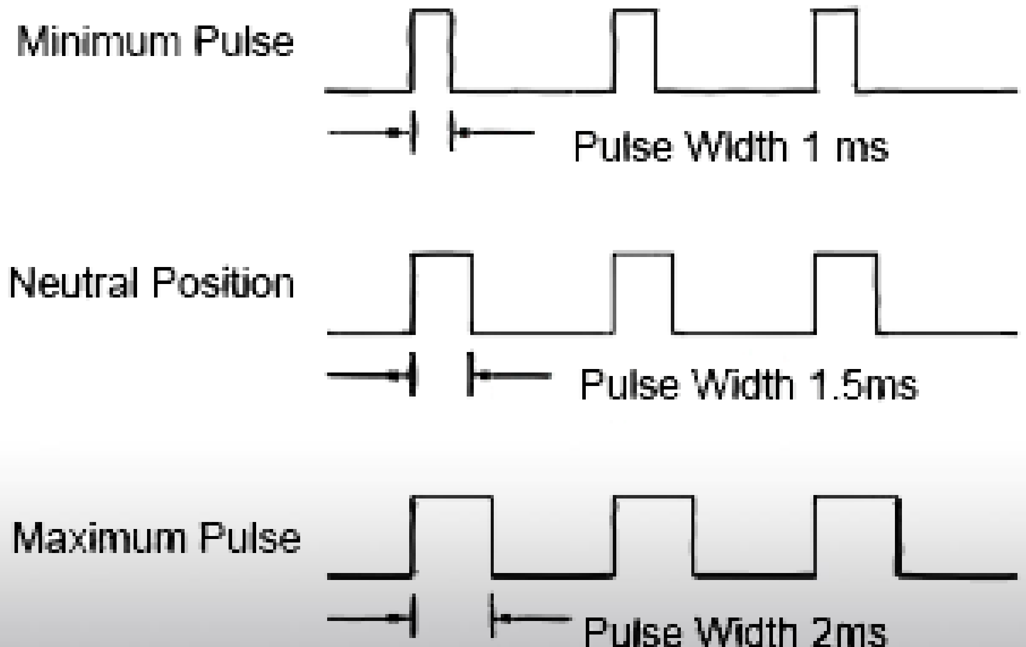
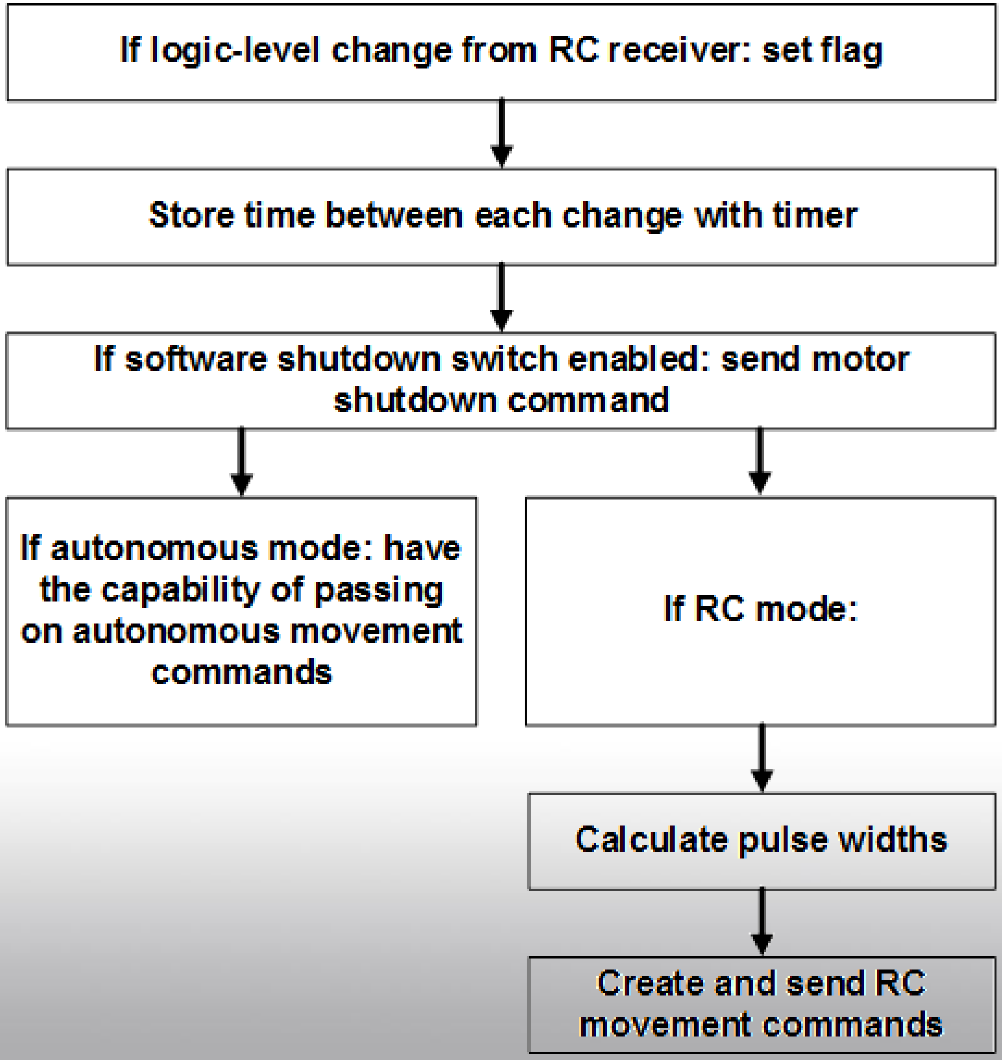


Image taken from [5].

Software Flowchart



Motor Movement Commands

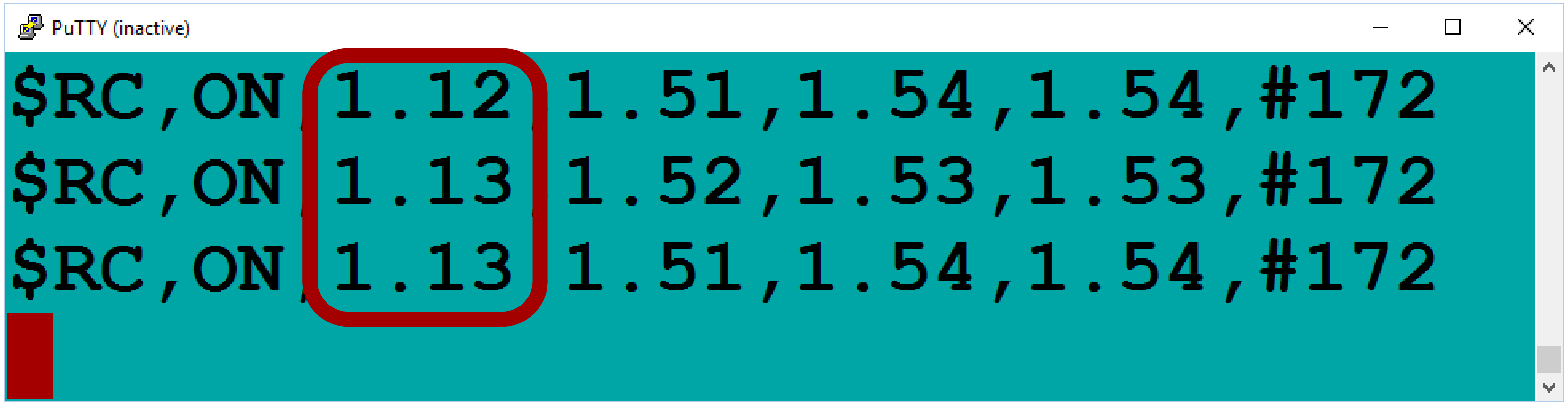
PuTTY (inactive)

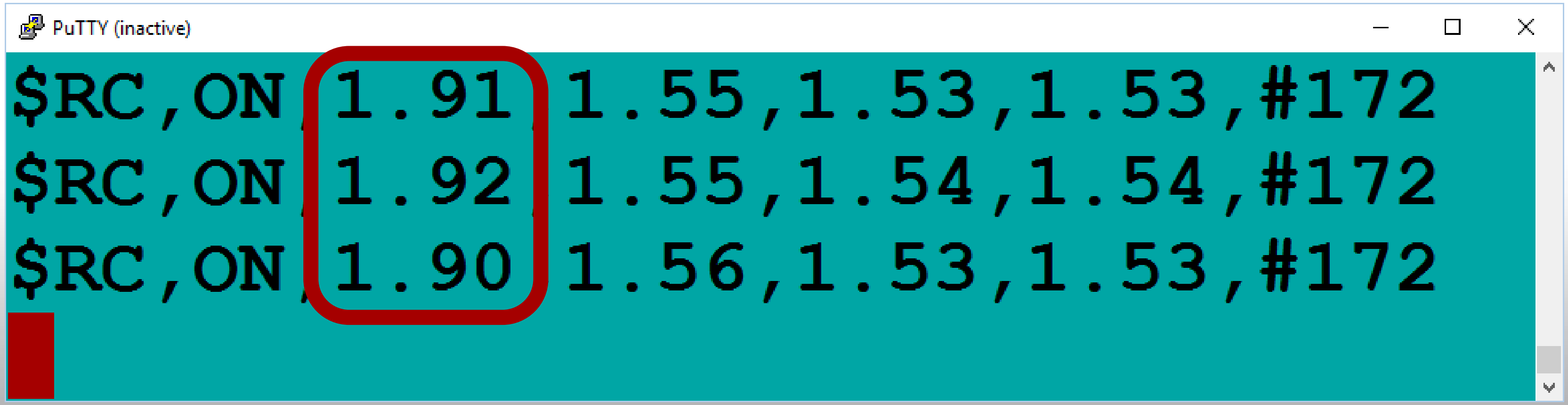
```
$RC,ON,1.12,1.51,1.54,1.54,#172
$RC,ON,1.13,1.52,1.53,1.53,#172
$RC,ON,1.13,1.51,1.54,1.54,#172
```

PuTTY (inactive)

```
$RC,ON,1.91,1.55,1.53,1.53,#172
$RC,ON,1.92,1.55,1.54,1.54,#172
$RC,ON,1.90,1.56,1.53,1.53,#172
```

Motor Movement Commands





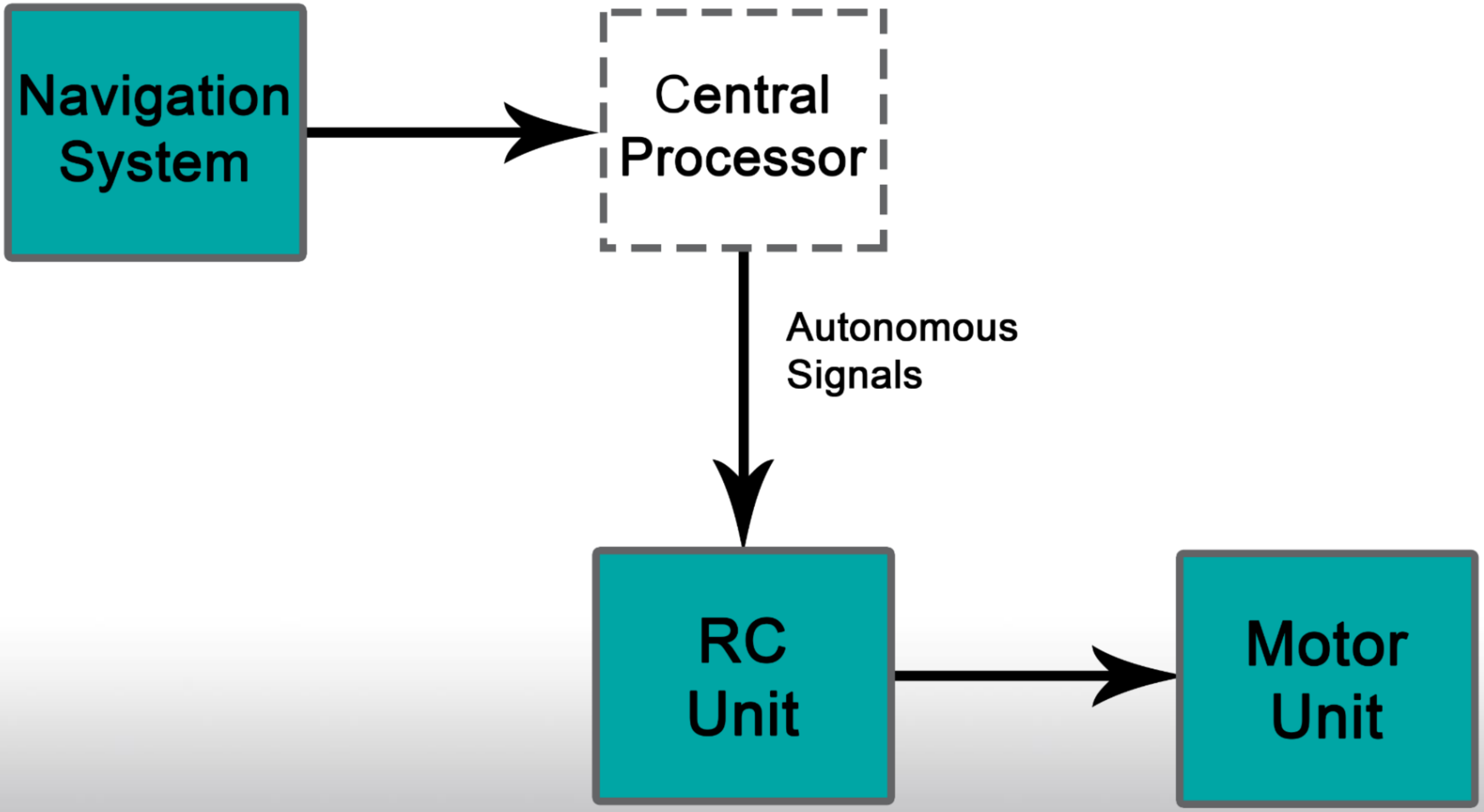
Presentation Outline

- Background
- Evan Dinelli
- Dan Van de Water
  - Motor Control Unit
    - Specifications
    - SPI Communication
    - Block Diagrams
- Michael Barnes
- Summary and Conclusions
- Questions and Answers (Q & A)

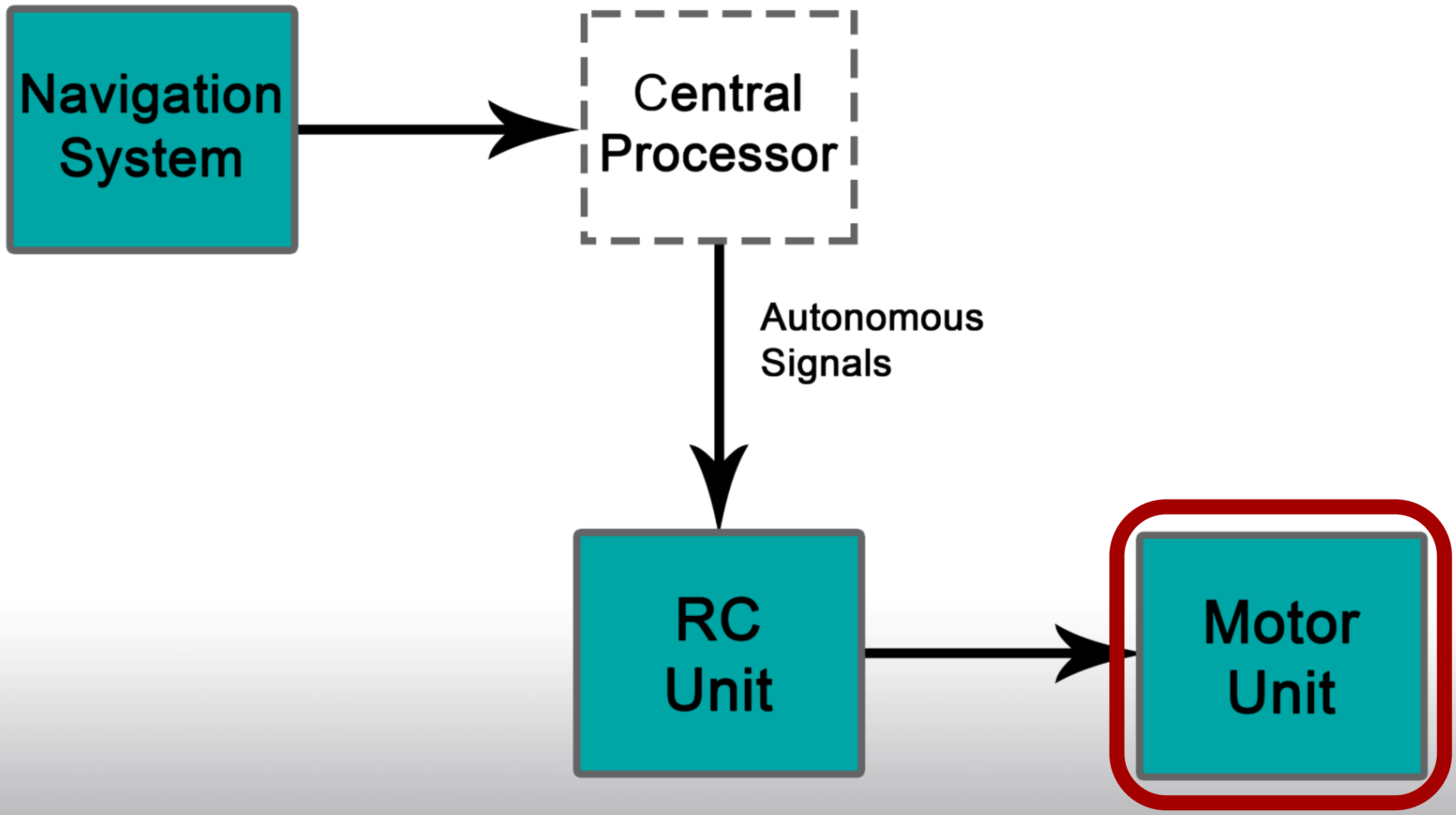
Functional Requirements

Functional Requirements	Specifications
GPS and Compass	GPS Accuracy: $< \pm 2$ m
	Compass Accuracy: $< \pm 2°$
Remote Control	Rate: ≥ 5 Commands / sec
	Mode Signal
	Software Kill Signal
Motor Control	Rate: ≥ 5 Commands / sec
	Physical Kill Switch

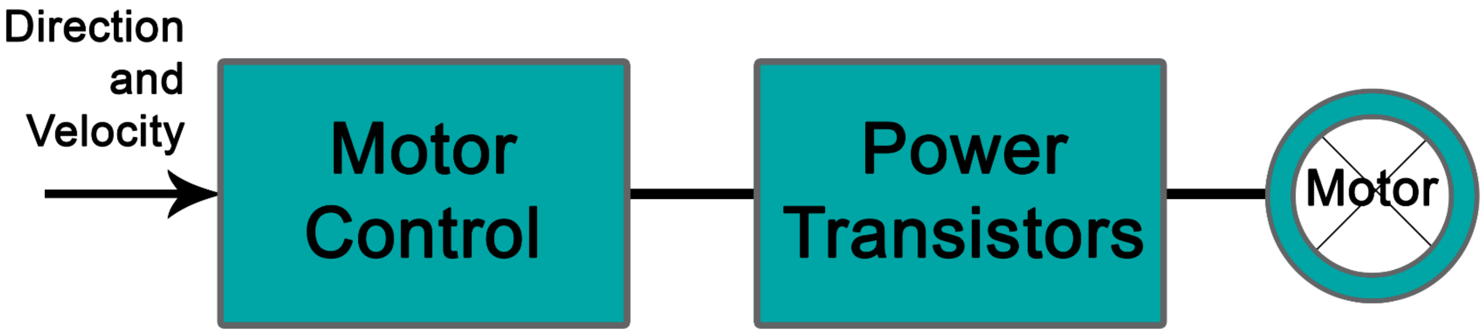
Block Diagram



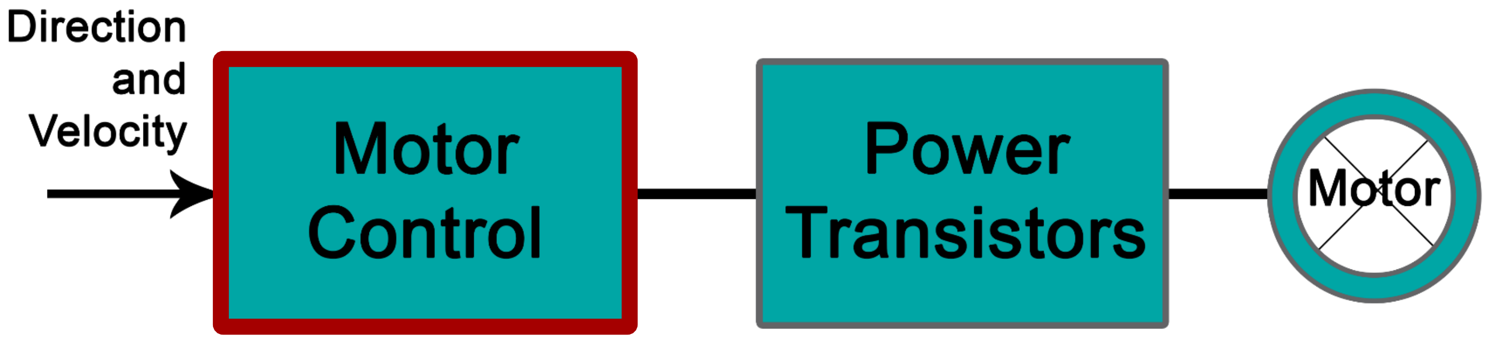
Block Diagram



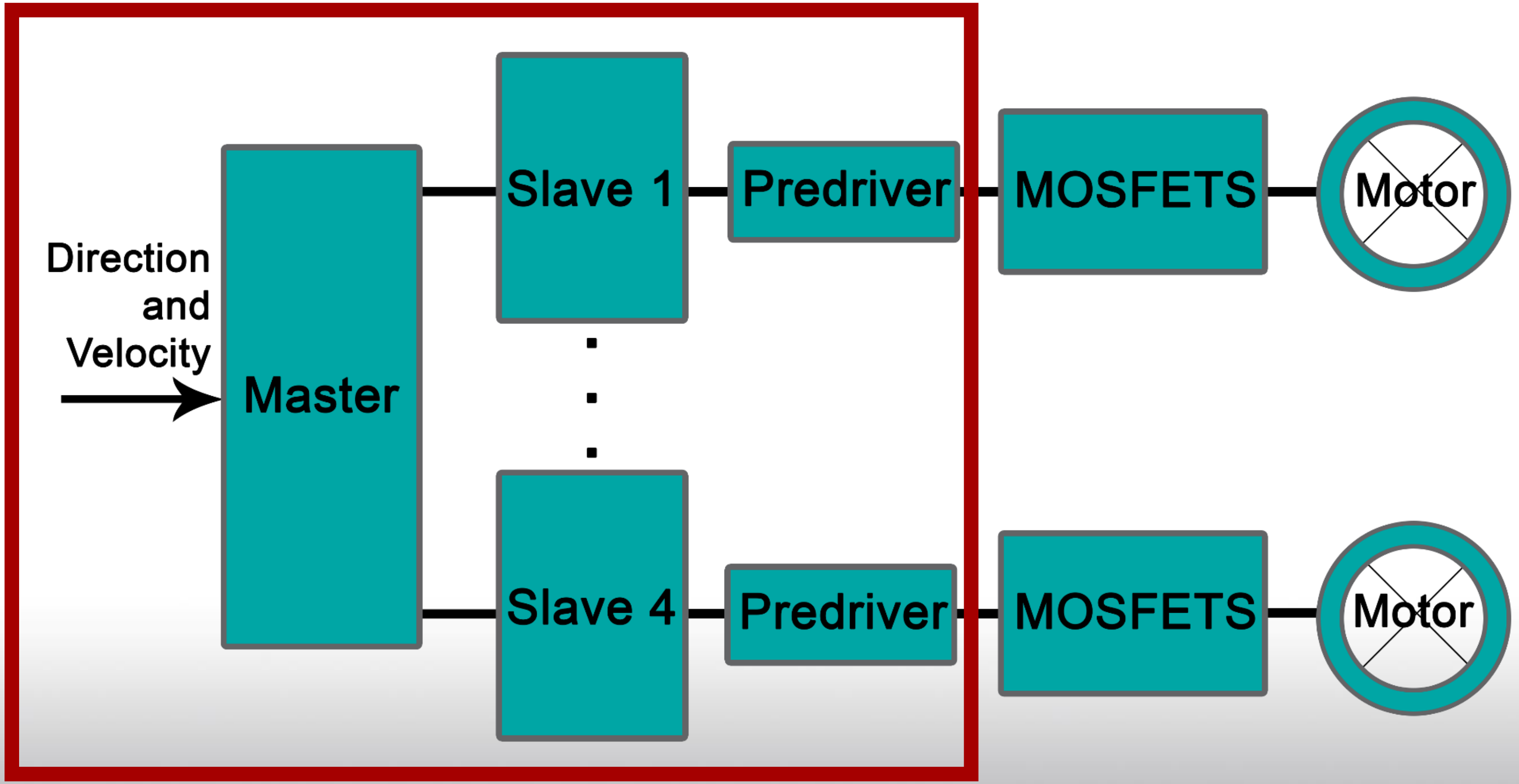
Block Diagram: Motor Unit



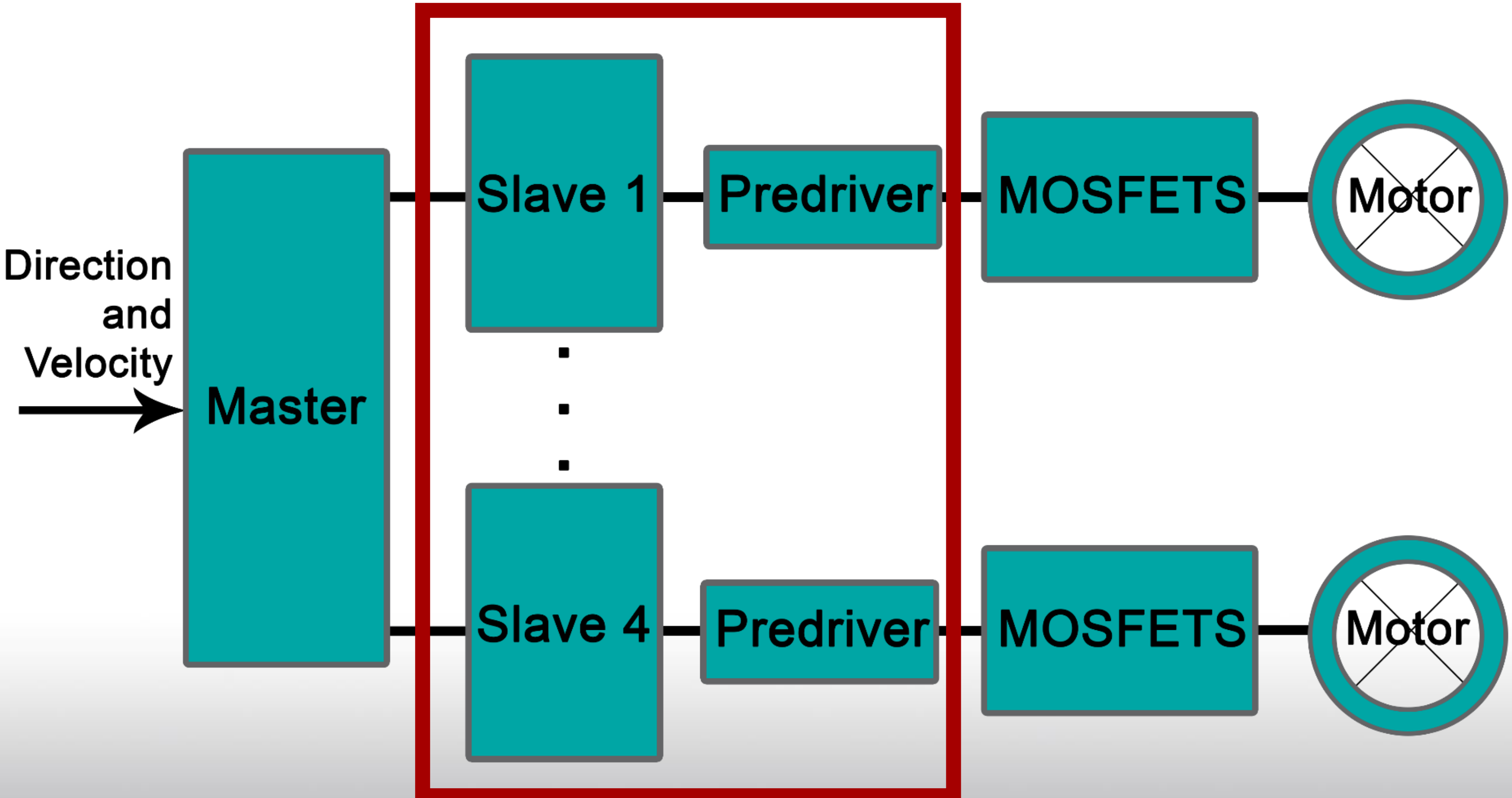
Block Diagram: Motor Unit



Block Diagram: Motor Control



Block Diagram: Motor Control



Predriver

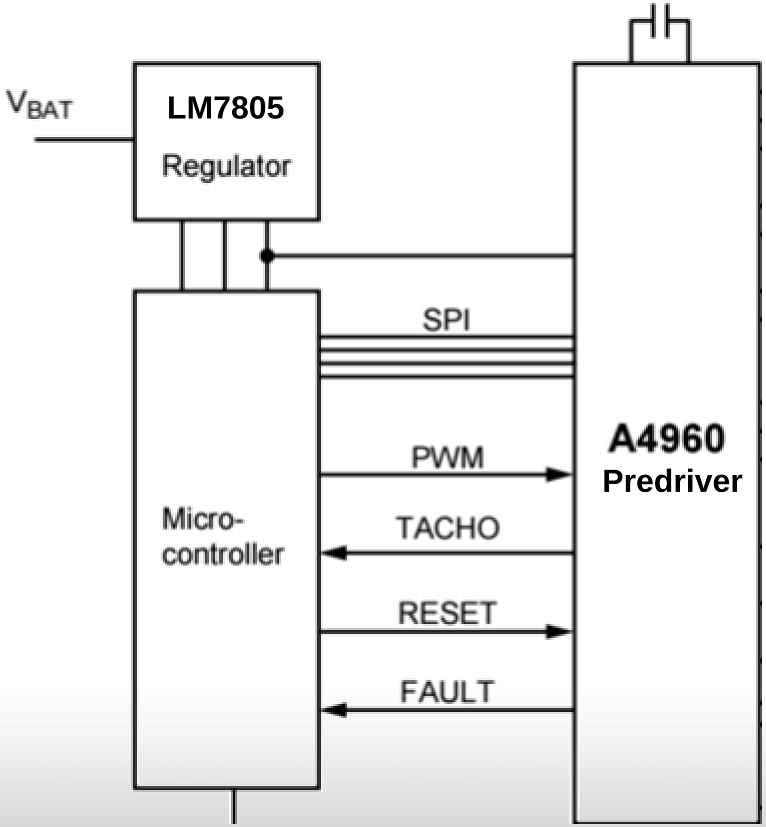


Image taken from [6].

Predriver

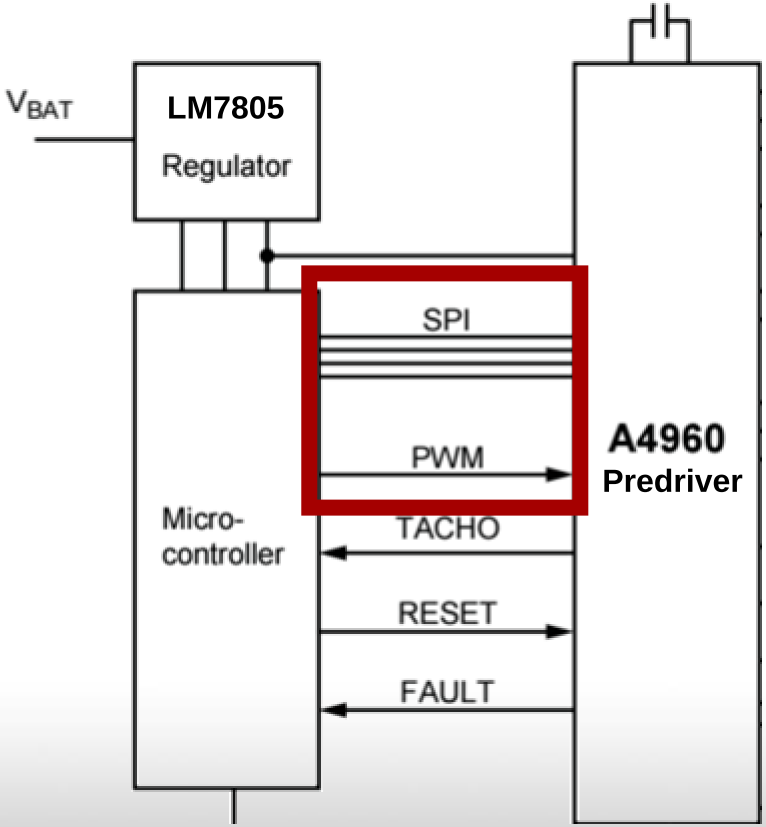
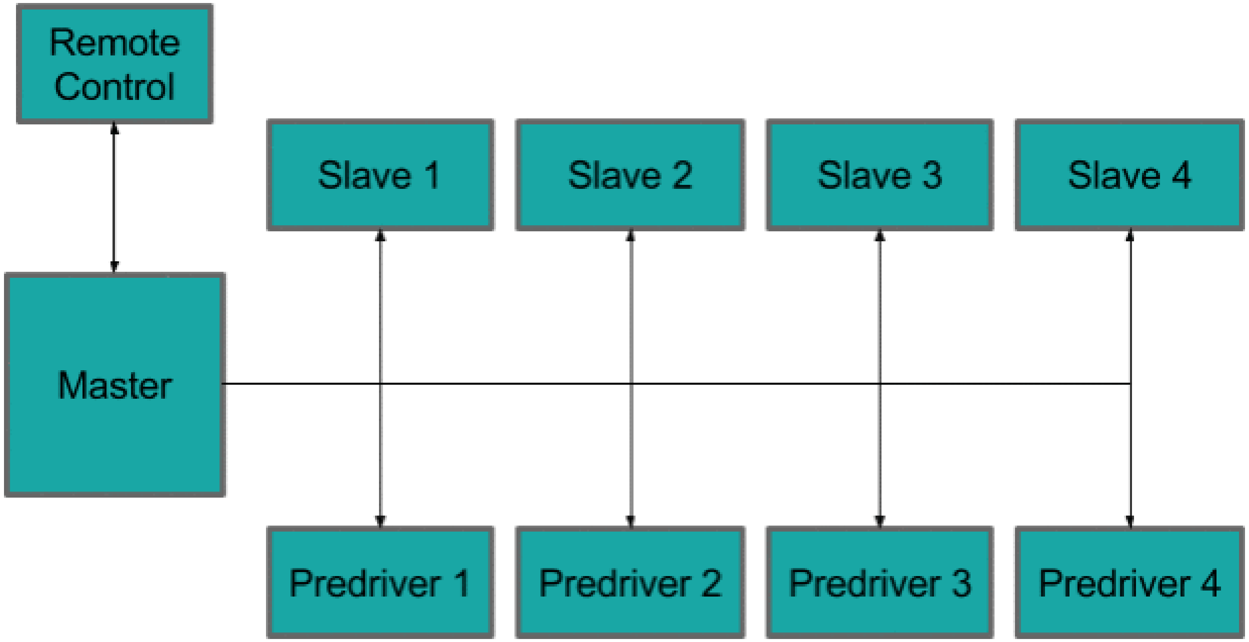
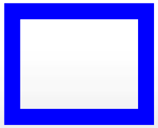


Image taken from [6].

Parallel SPI Communication

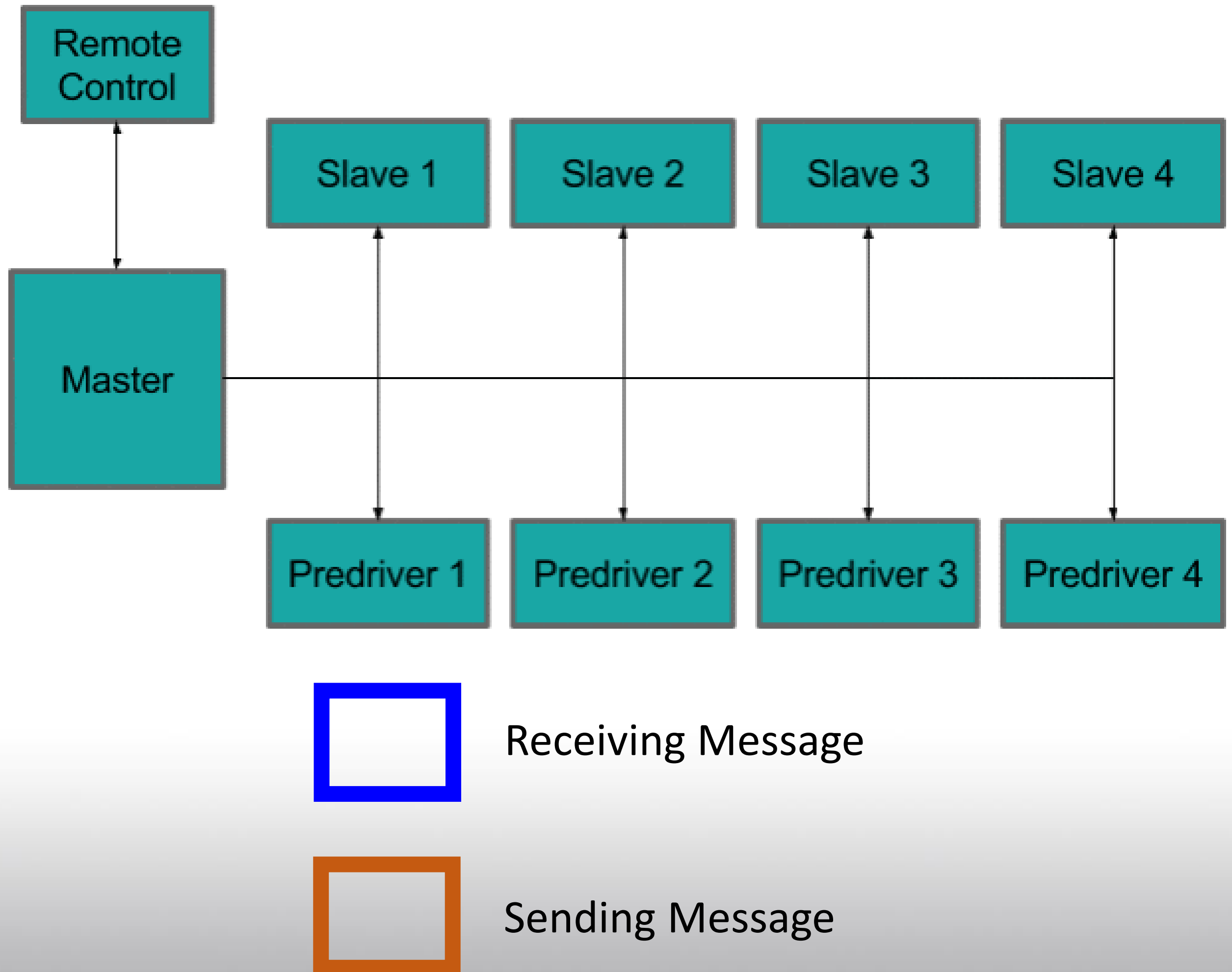




Receiving Message

Sending Message

Parallel SPI Communication



Interrupts

Interrupts	Priority	Function
Reset	1	Reset Microcontroller
Watchdog Timer	9	Reset
Timer 2	12	Reset Watchdog, PWM Generation
Timer 0	19	1 ms Interrupt
SPI	20	Serial Transfer Complete
USART RX	21	Receive Complete

Interrupts

Interrupts	Priority	Function
Reset	1	Reset Microcontroller
Watchdog Timer	9	Reset
Timer 2	12	Reset Watchdog, PWM Generation
Timer 0	19	1 ms Interrupt
SPI	20	Serial Transfer Complete
USART RX	21	Receive Complete

Interrupts

Interrupts	Priority	Function
Reset	1	Reset Microcontroller
Watchdog Timer	9	Reset
Timer 2	12	Reset Watchdog, PWM Generation
Timer 0	19	1 ms Interrupt
SPI	20	Serial Transfer Complete
USART RX	21	Receive Complete

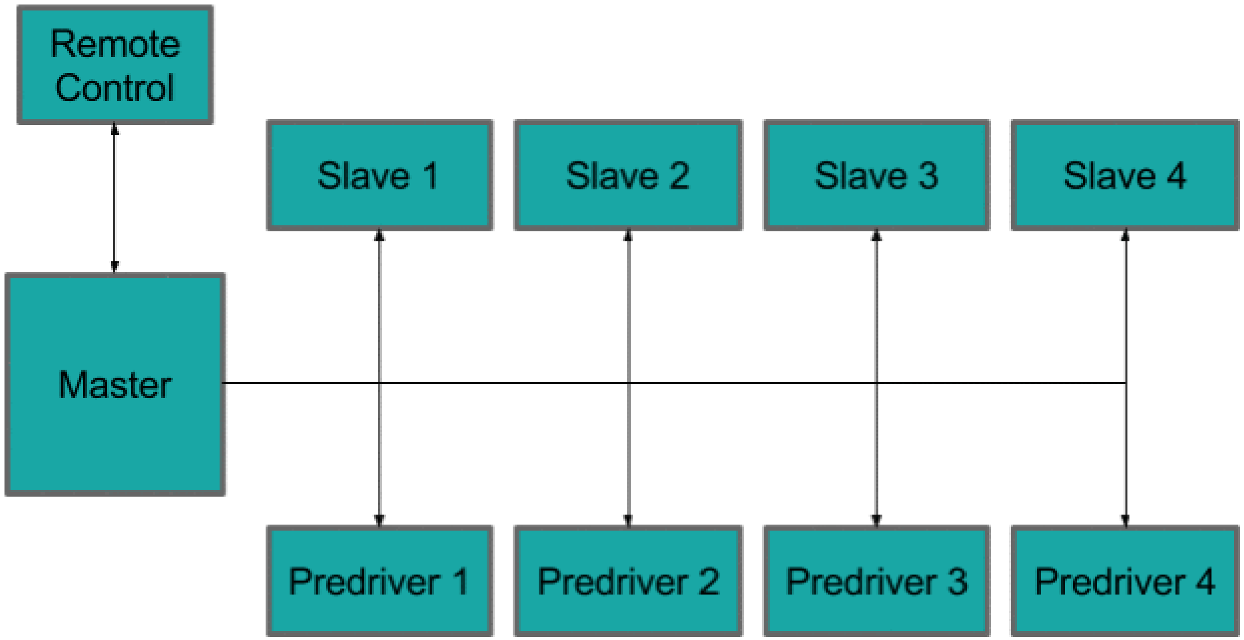
Interrupts

Interrupts	Priority	Function
Reset	1	Reset Microcontroller
Watchdog Timer	9	Reset
Timer 2	12	Reset Watchdog, PWM Generation
Timer 0	19	1 ms Interrupt
SPI	20	Serial Transfer Complete
USART RX	21	Receive Complete

Interrupts

Interrupts	Priority	Function
Reset	1	Reset Microcontroller
Watchdog Timer	9	Reset
Timer 2	12	Reset Watchdog, PWM Generation
Timer 0	19	1 ms Interrupt
SPI	20	Serial Transfer Complete
USART RX	21	Receive Complete

Parallel SPI Communication

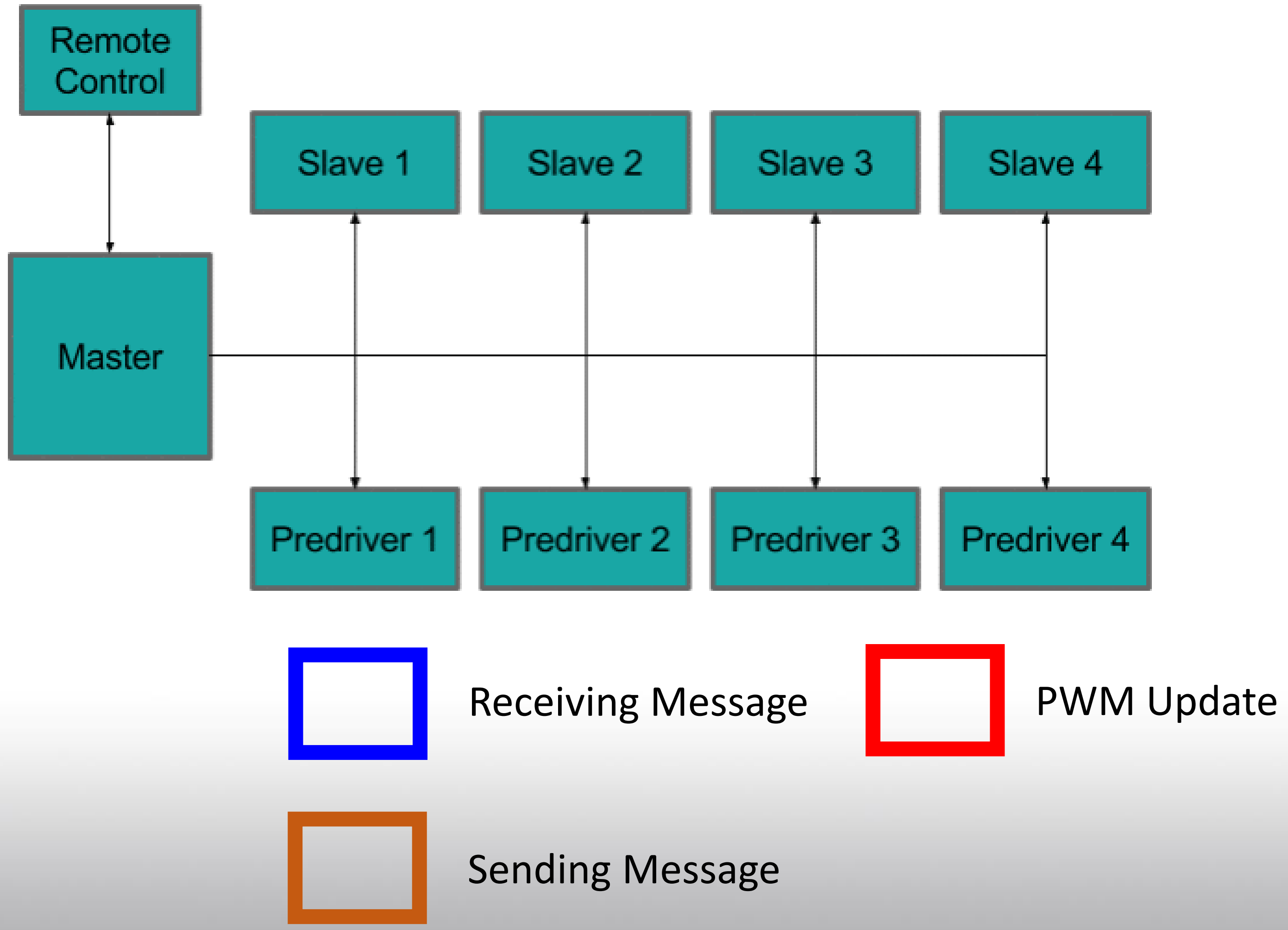


Receiving Message

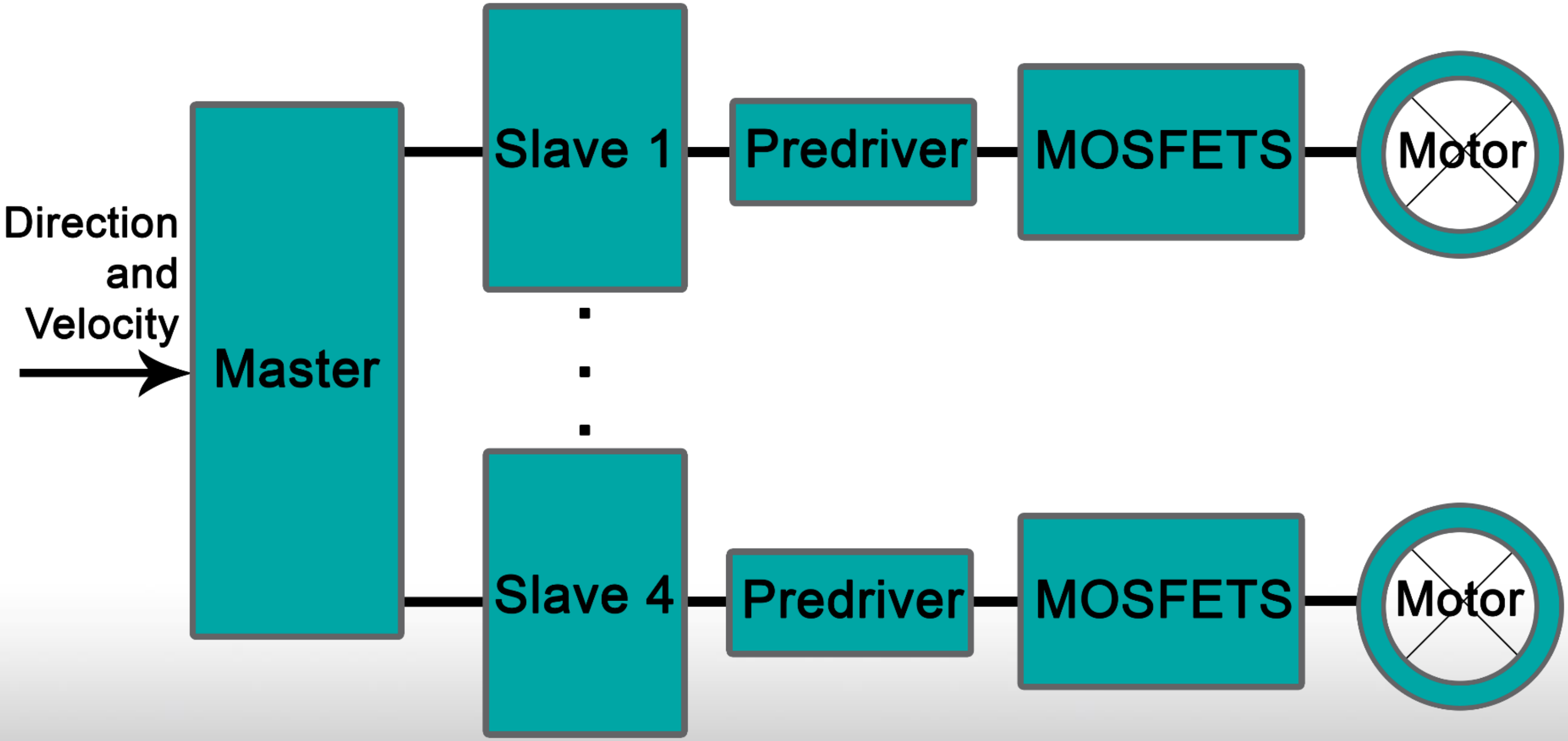
PWM Update

Sending Message

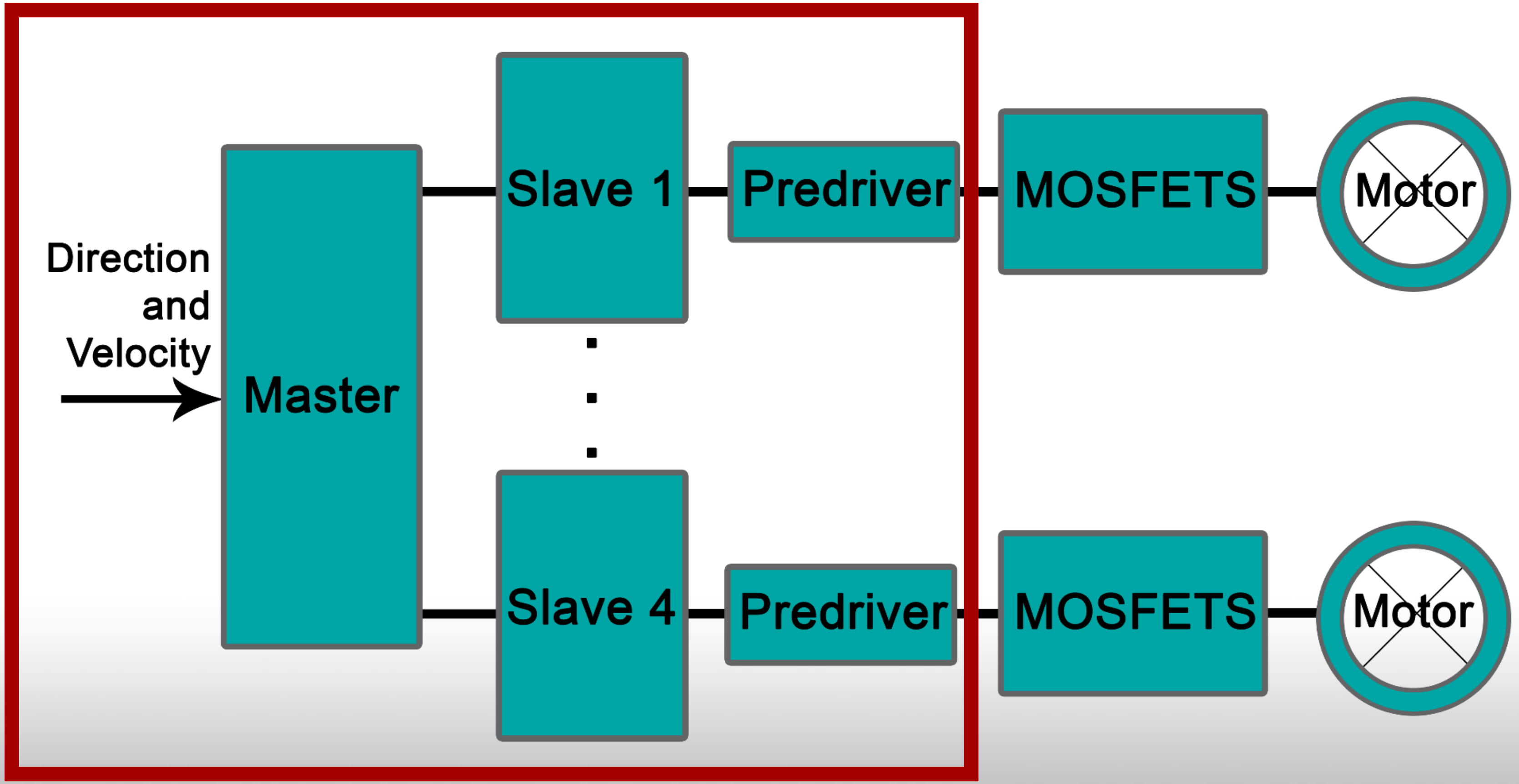
Parallel SPI Communication



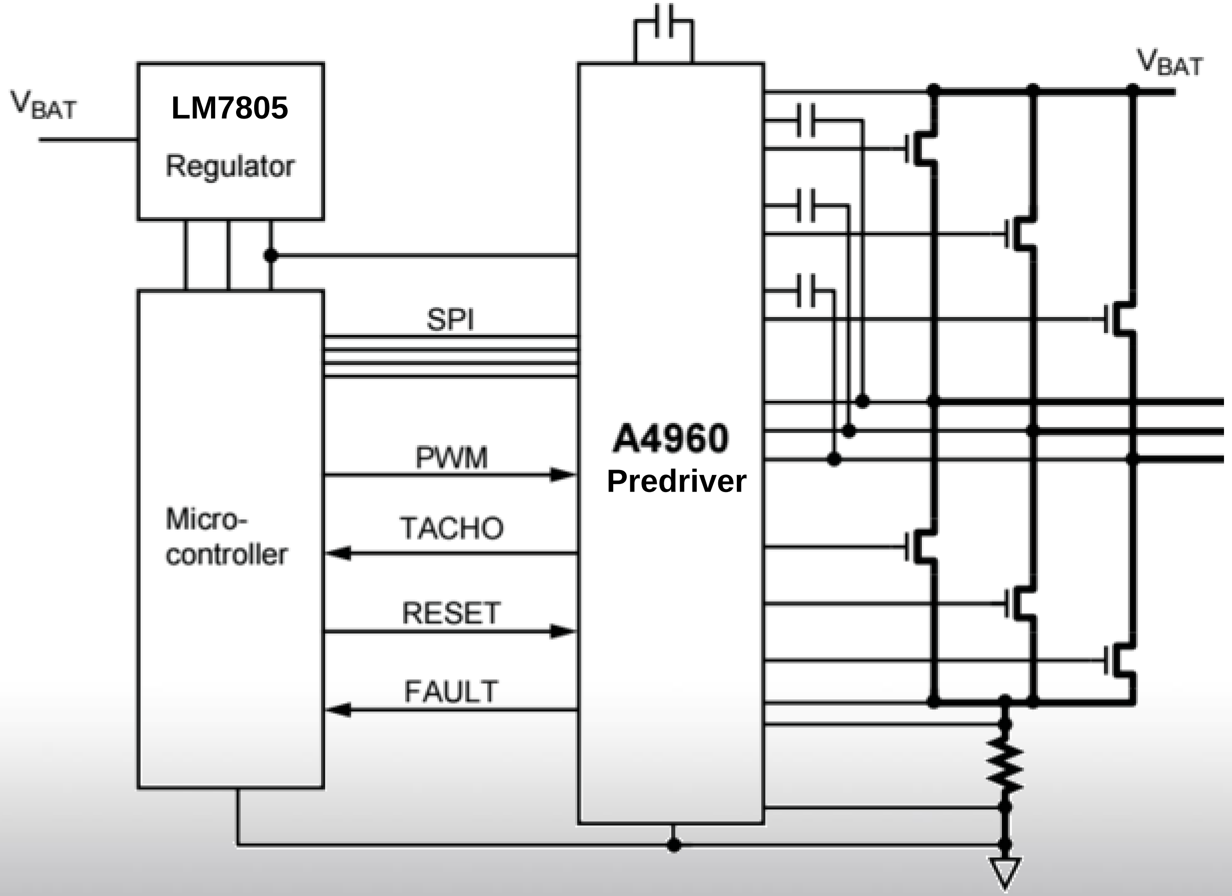
Motor Control Unit



Block Diagram: Motor Control

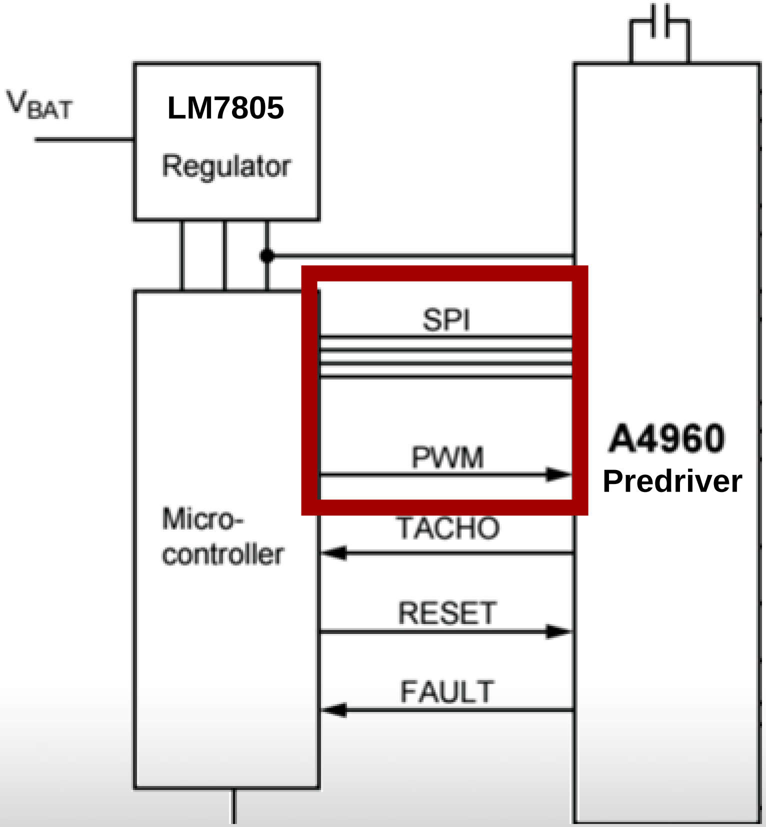


Predriver



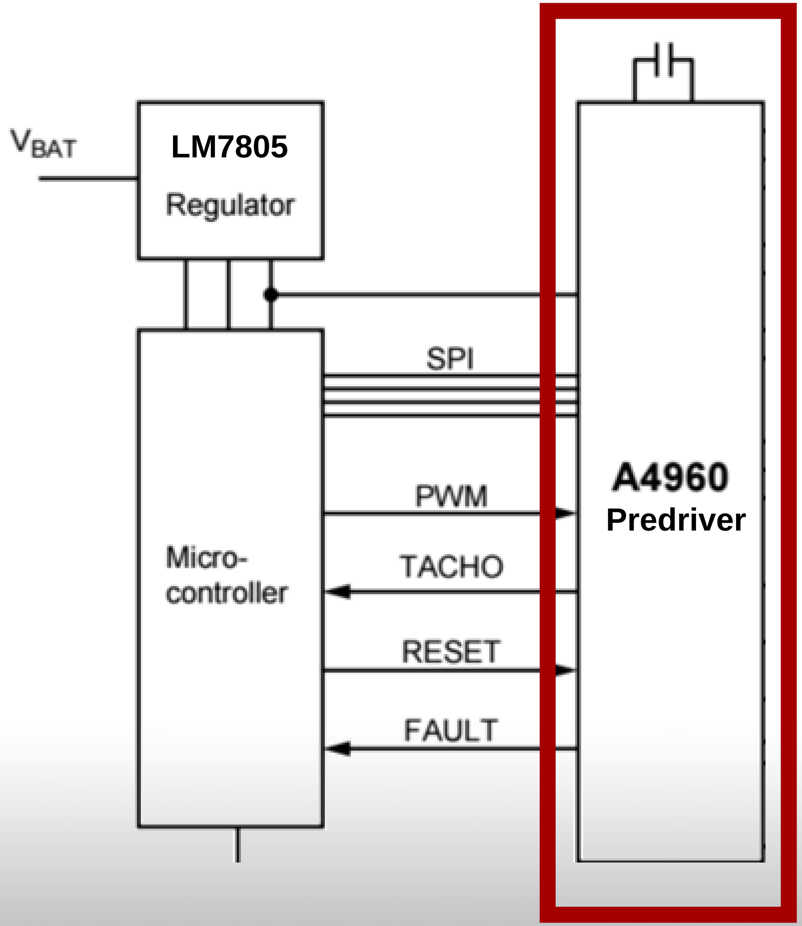
VBAT
LM7805
Regulator
SPI
PWM
TACHO
RESET
FAULT
Micro-
controller
A4960
Predriver
VBAT

Predriver



V_{BAT}
LM7805
Regulator
SPI
PWM
TACHO
RESET
FAULT
Micro-
controller
A4960
Predriver

Predriver



V_{BAT}
LM7805
Regulator
SPI
PWM
TACHO
RESET
FAULT
Micro-
controller
A4960
Predriver

Predriver

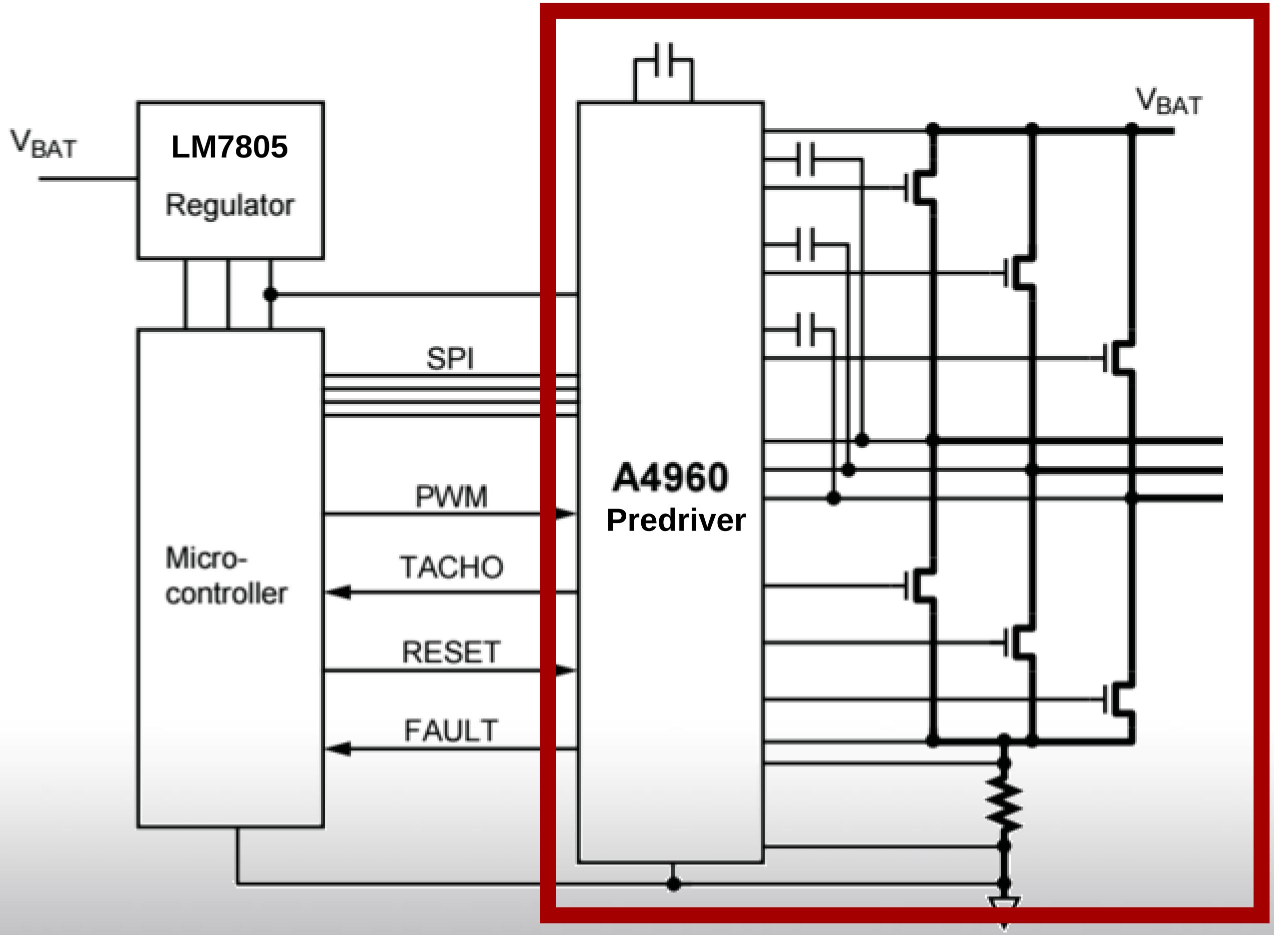
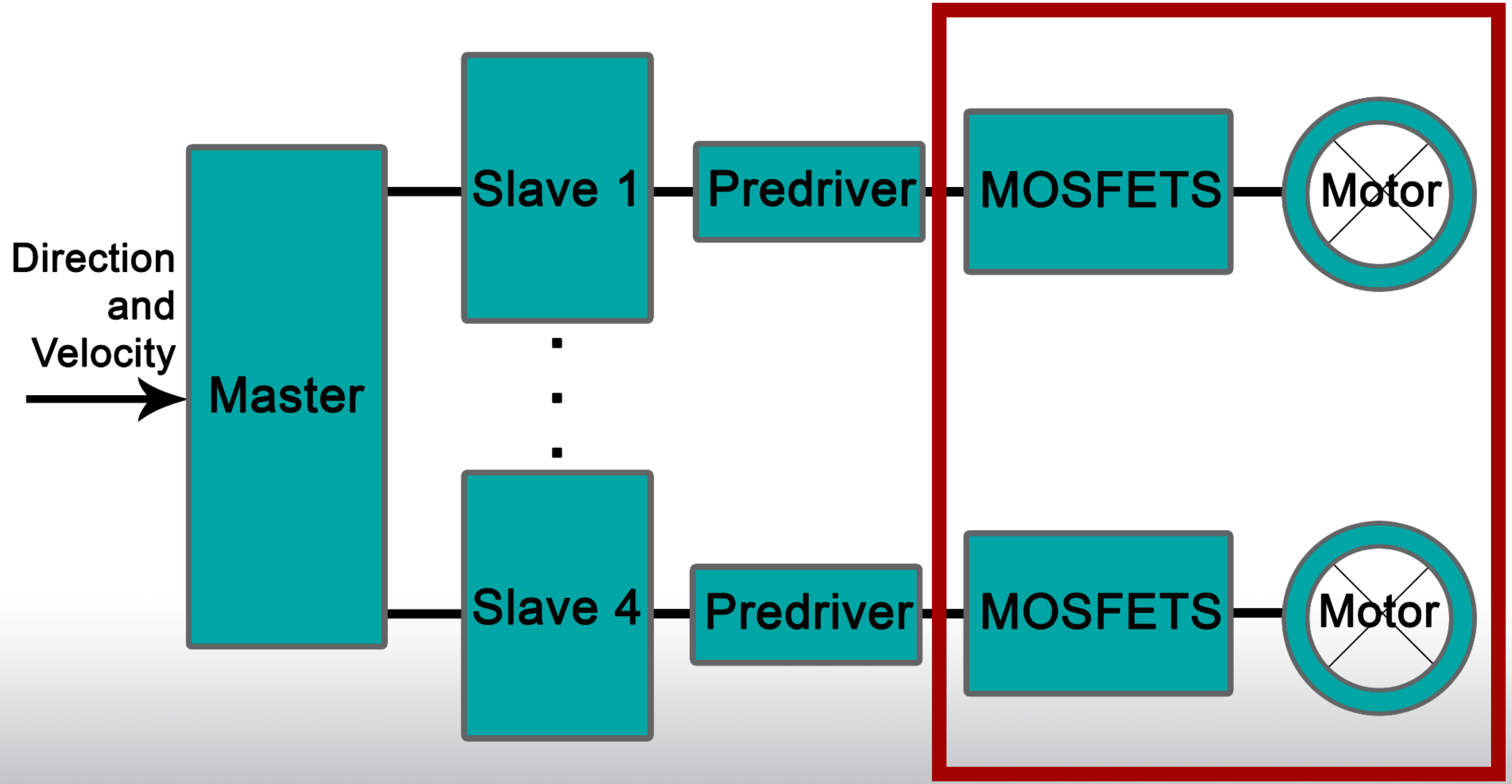


Image taken from [6].

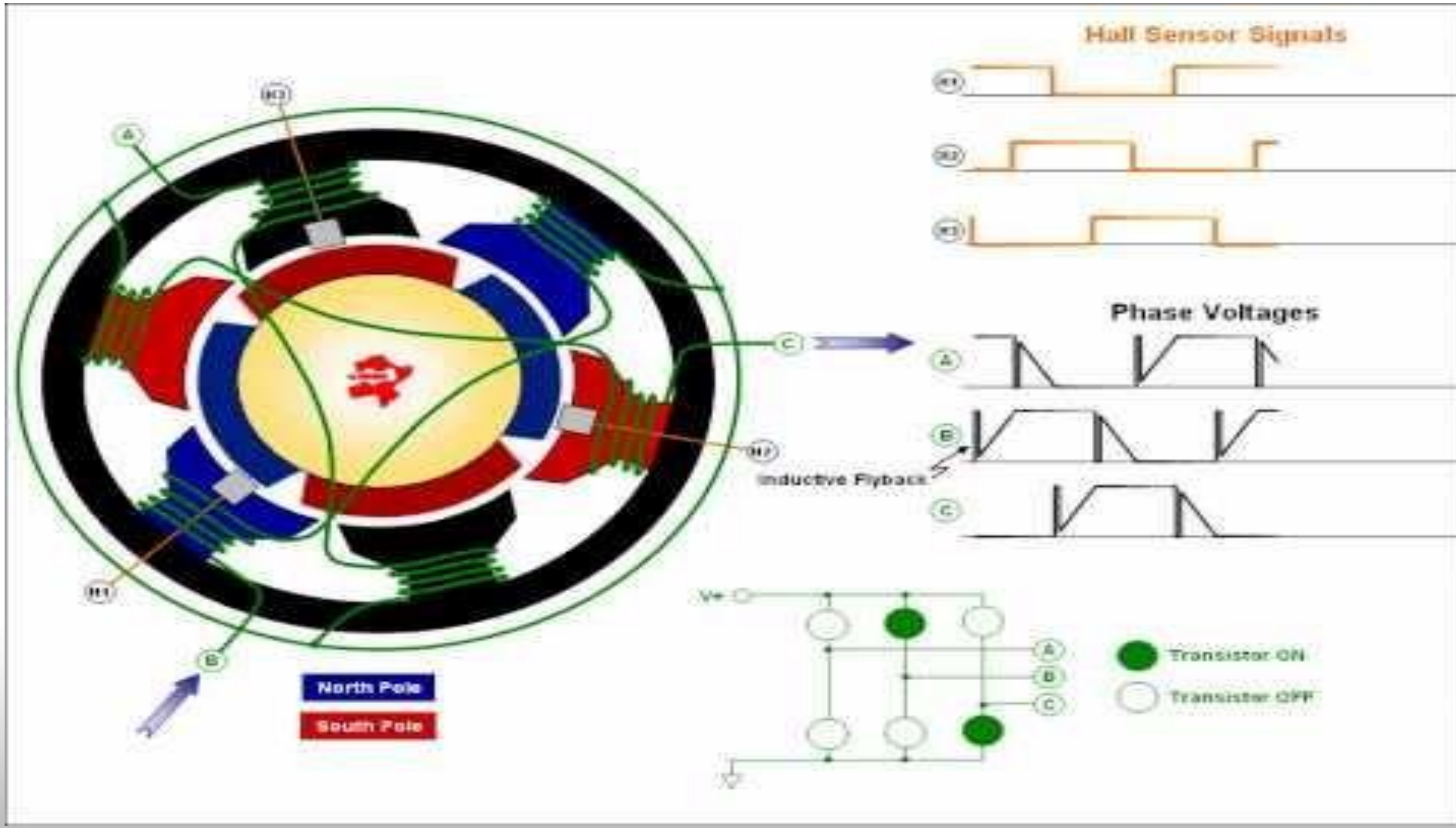
Presentation Outline

- Background
- Evan Dinelli
- Dan Van de Water
- Michael Barnes
  - Power Transistors
- Summary and Conclusions
- Questions and Answers (Q & A)

Motor Unit

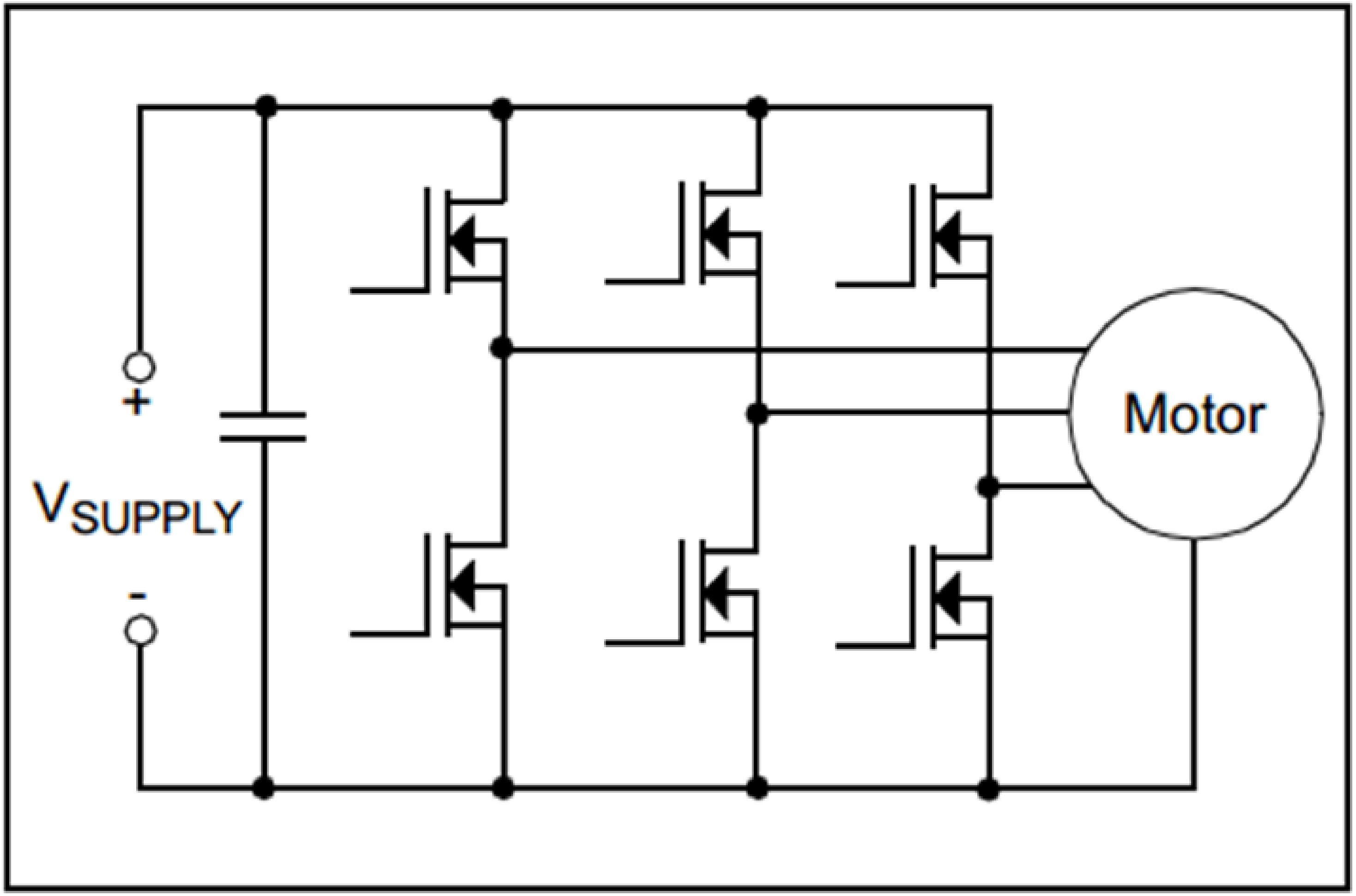

Direction
and
Velocity
Master
Slave 1
Slave 4
Predriver
Predriver
MOSFETS
MOSFETS
Motor
Motor

BLDC Motors

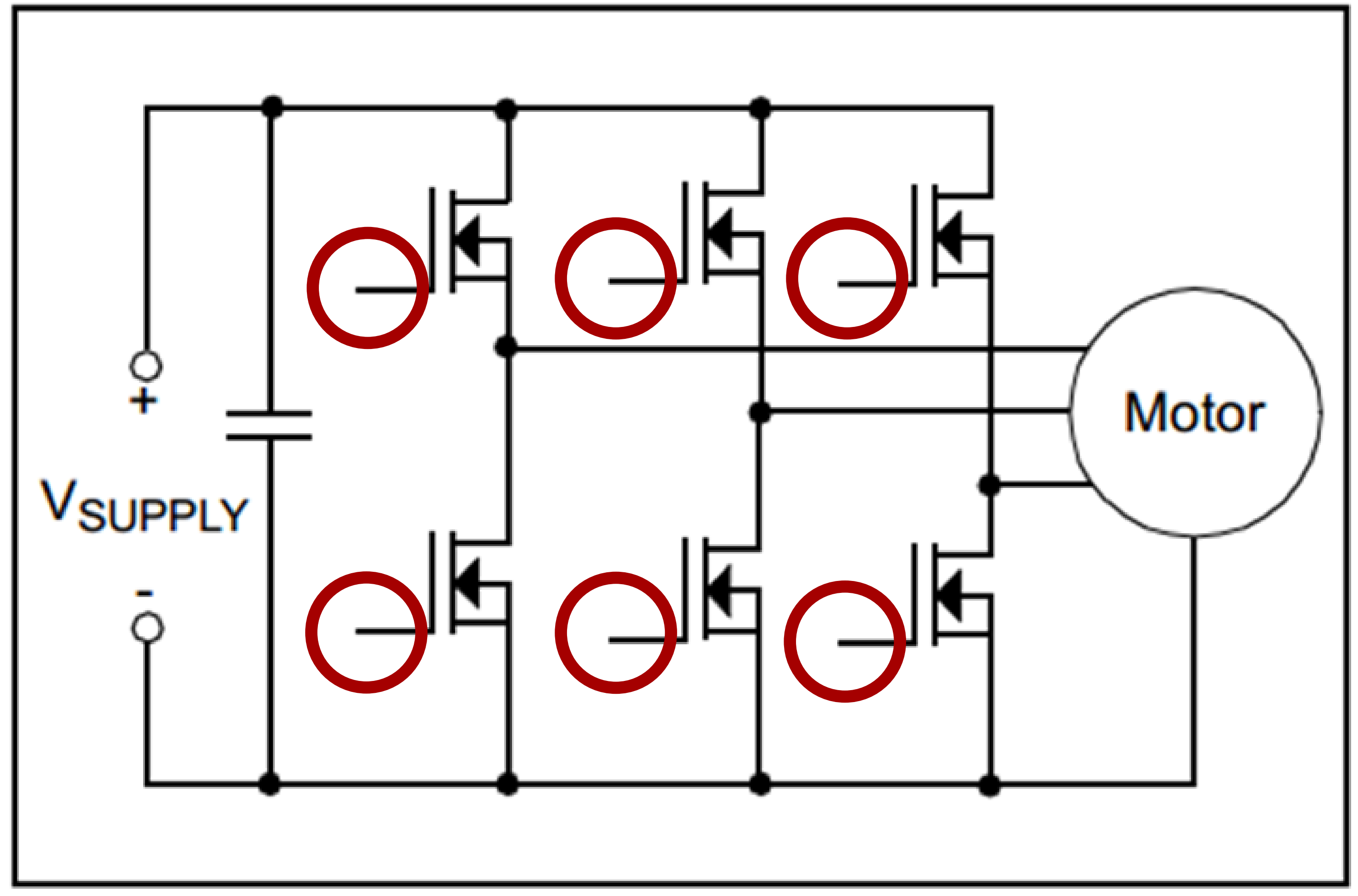


Video taken from [7].

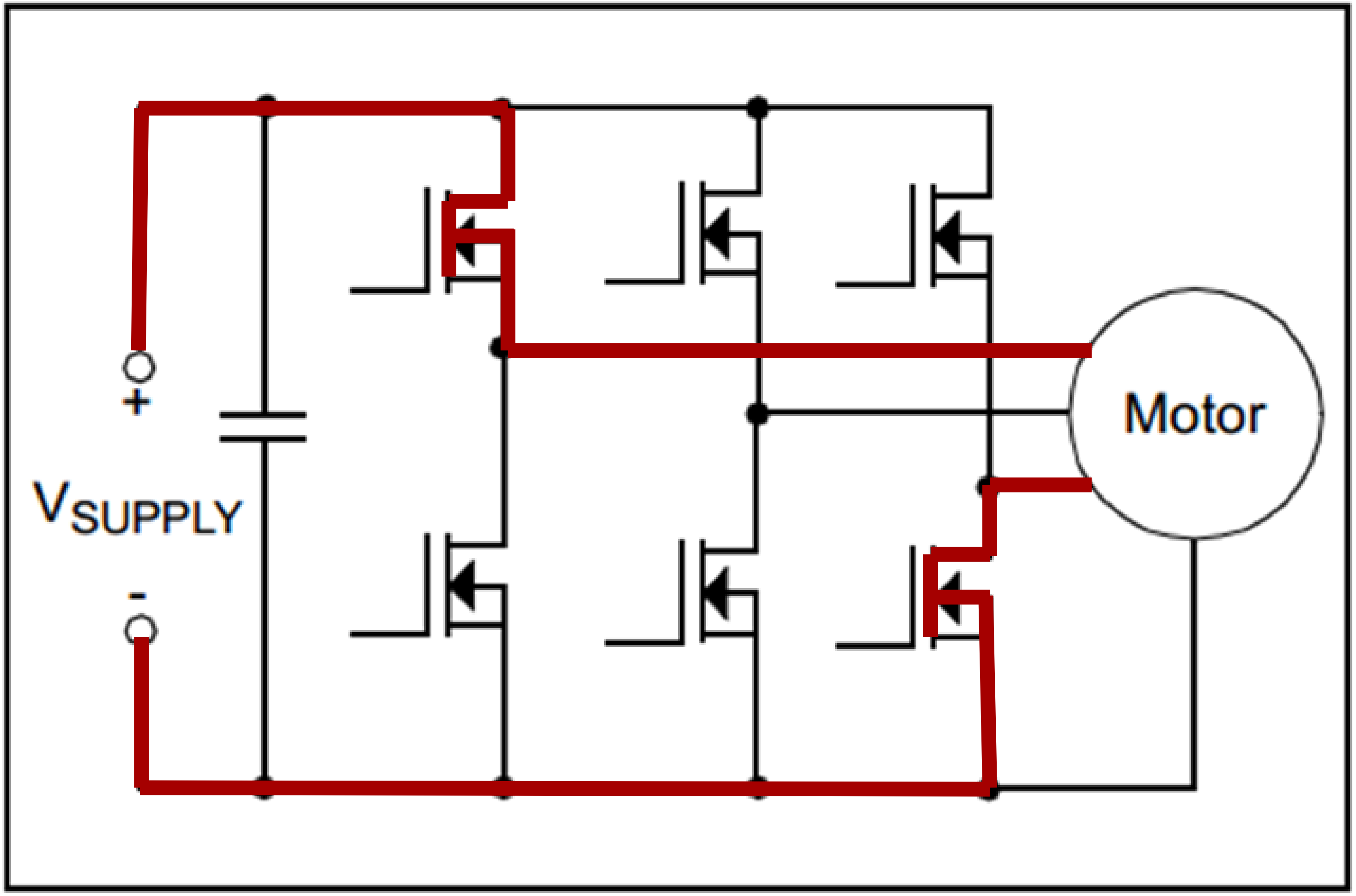
BLDC – MOSFET Schematic



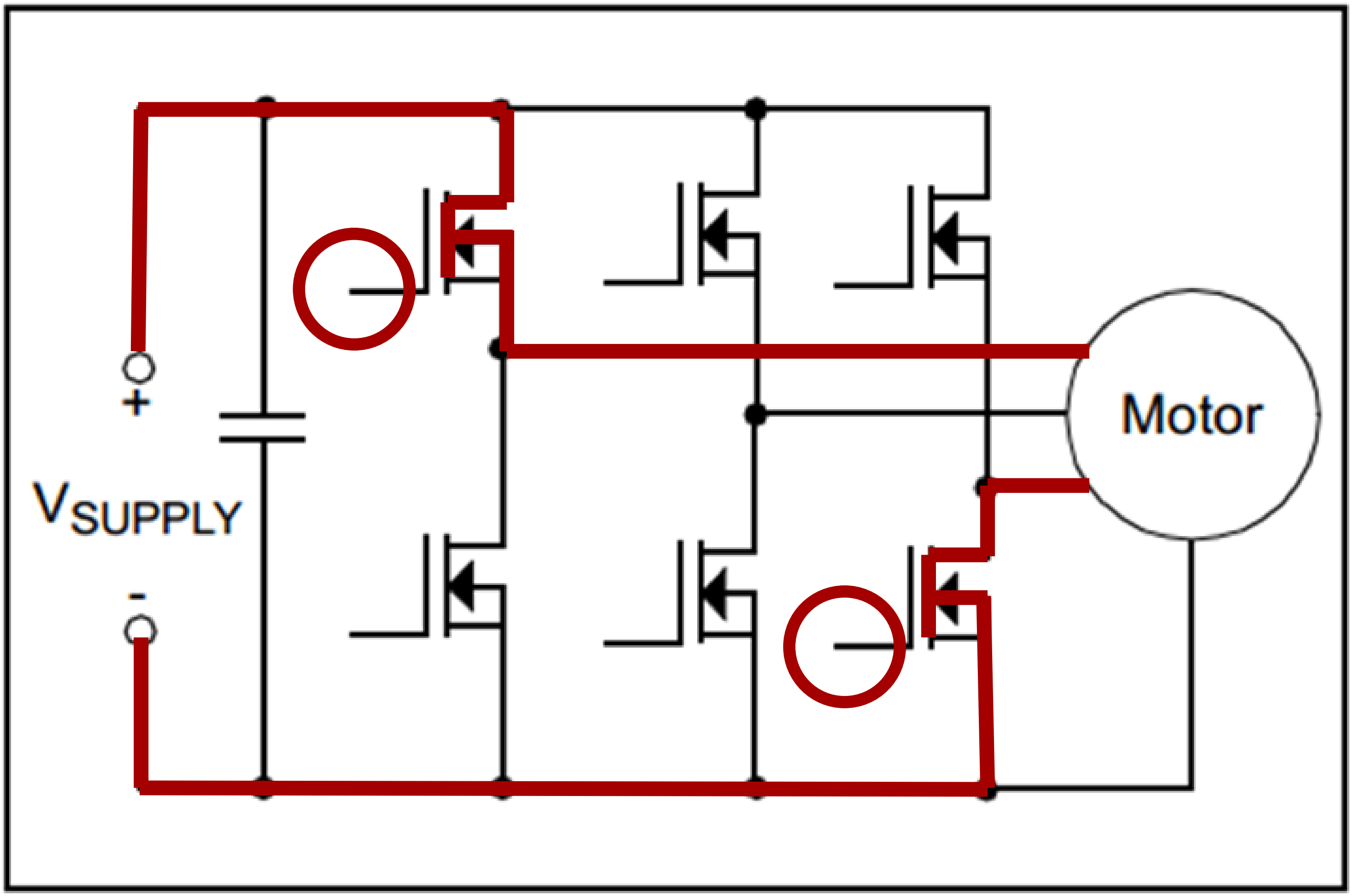
BLDC – MOSFET Schematic



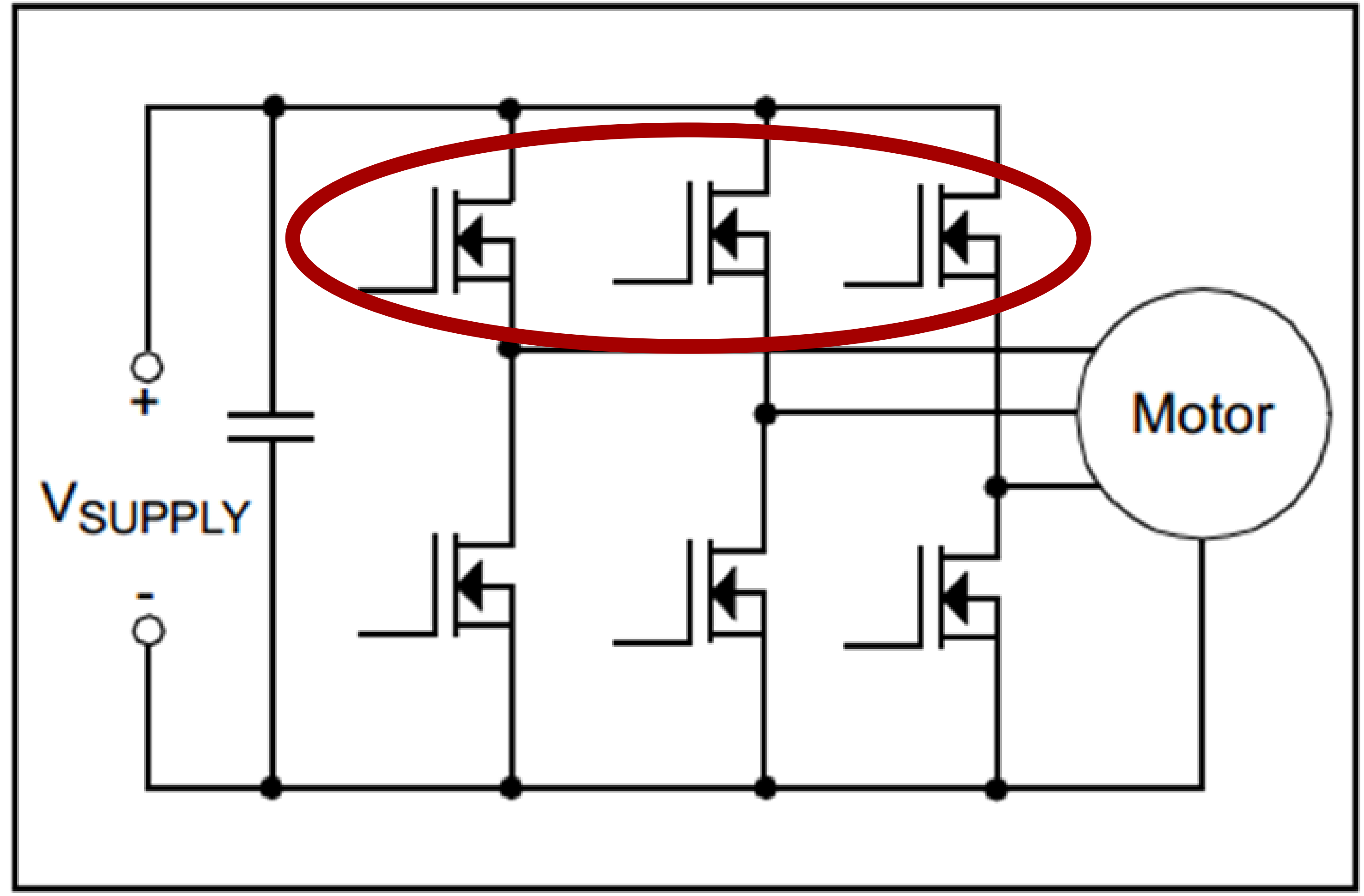
BLDC – MOSFET Schematic



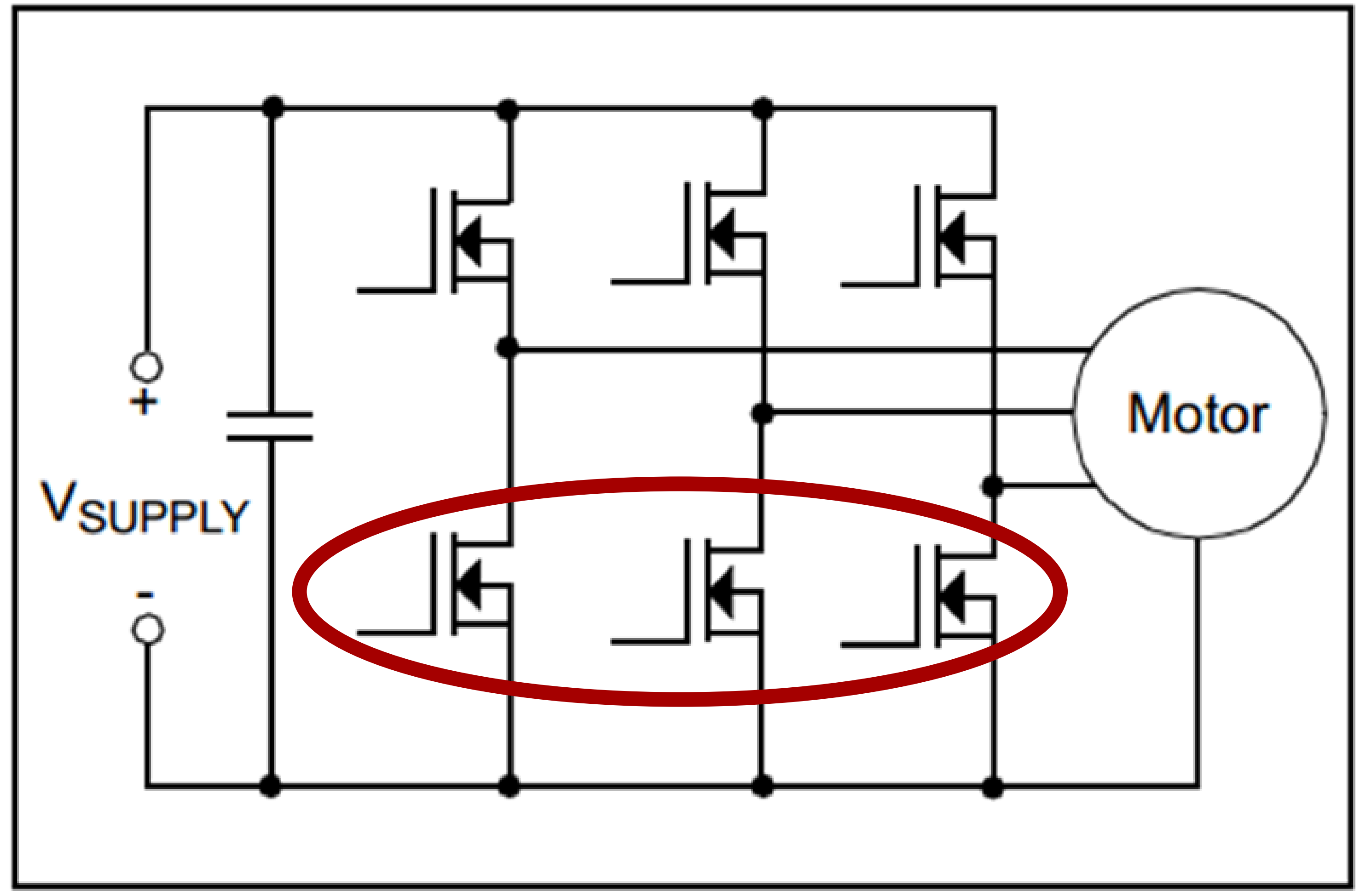
BLDC – MOSFET Schematic



BLDC – MOSFET Schematic



BLDC – MOSFET Schematic



N-Channel MOSFETs

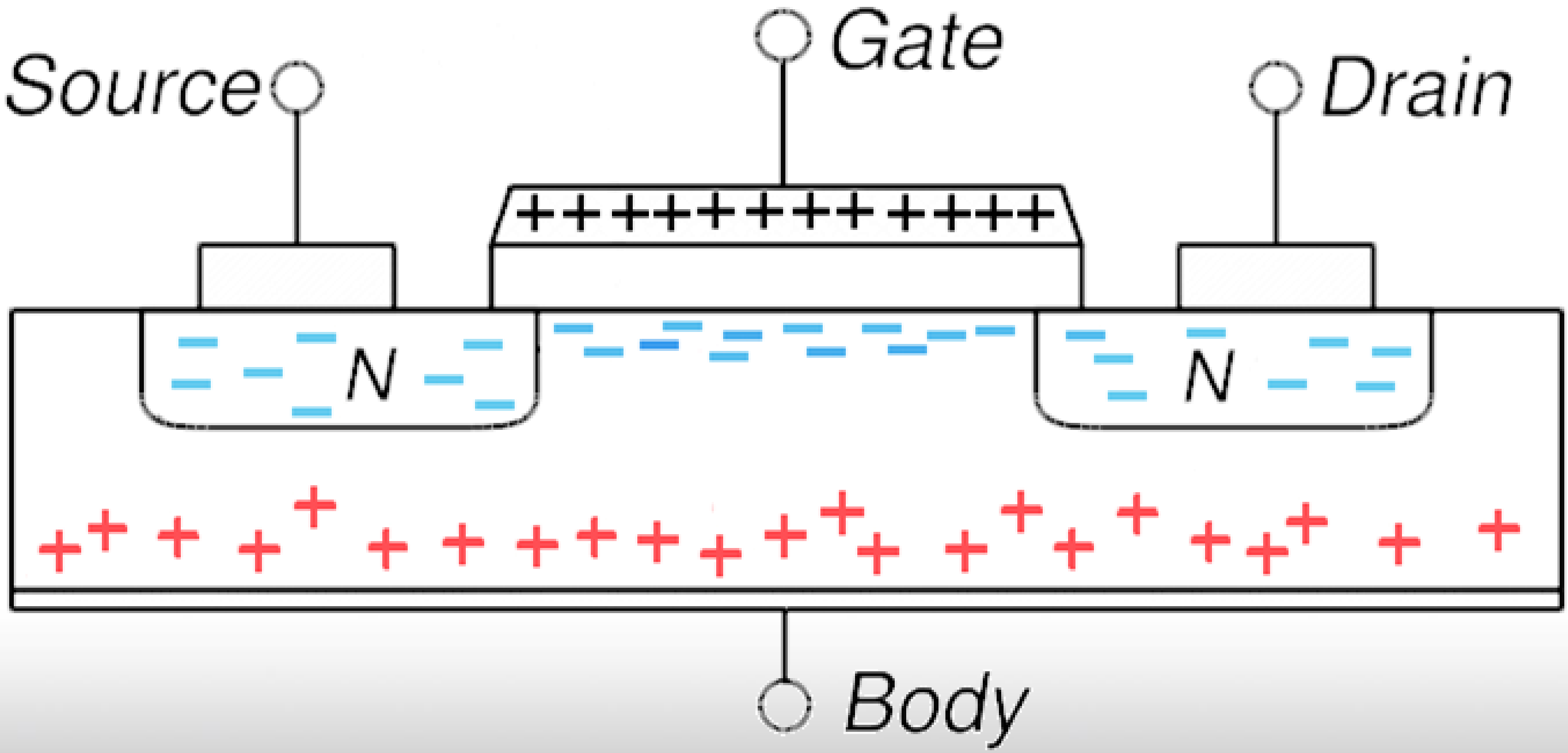


Image taken from [10].

N-Channel MOSFETs

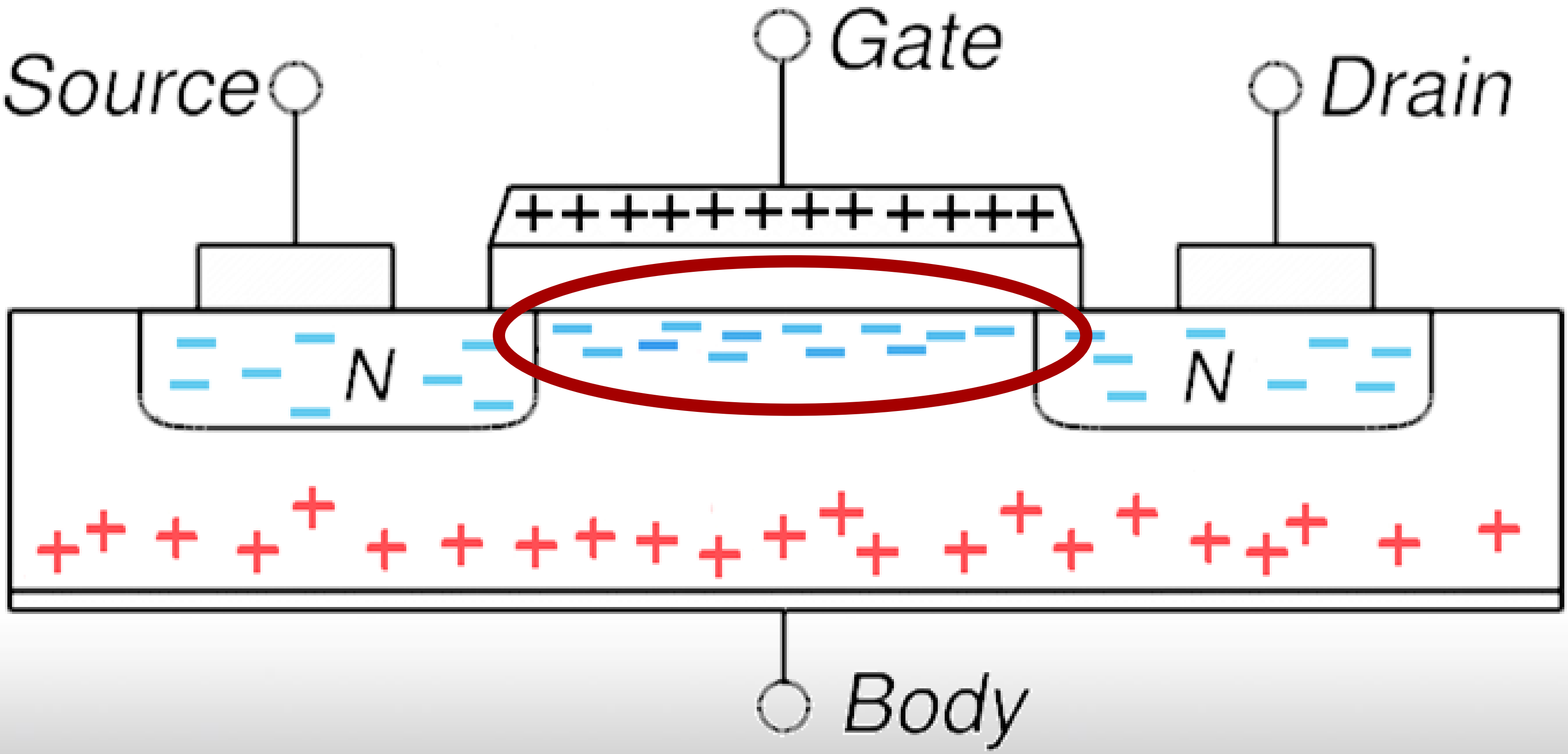
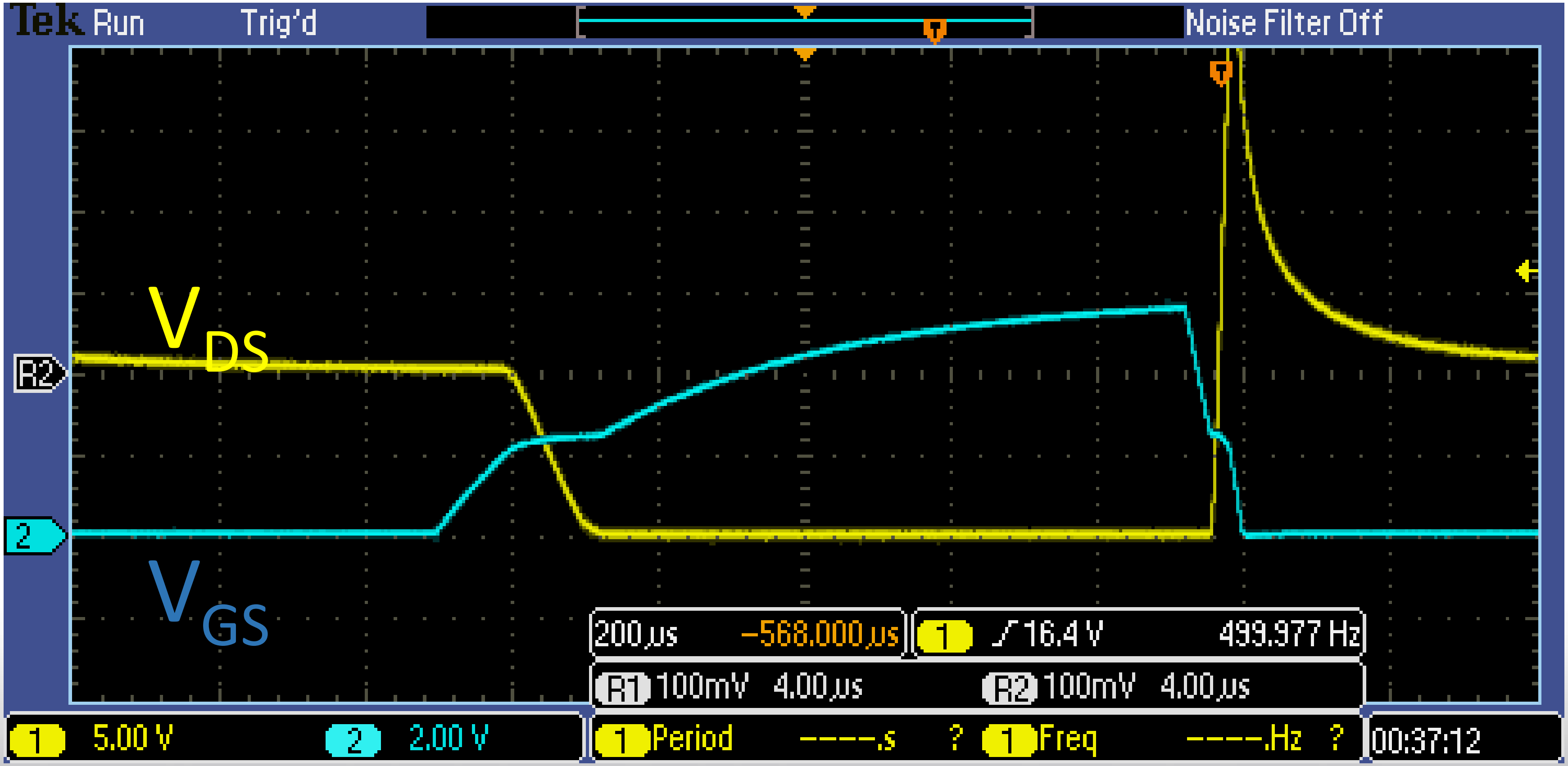


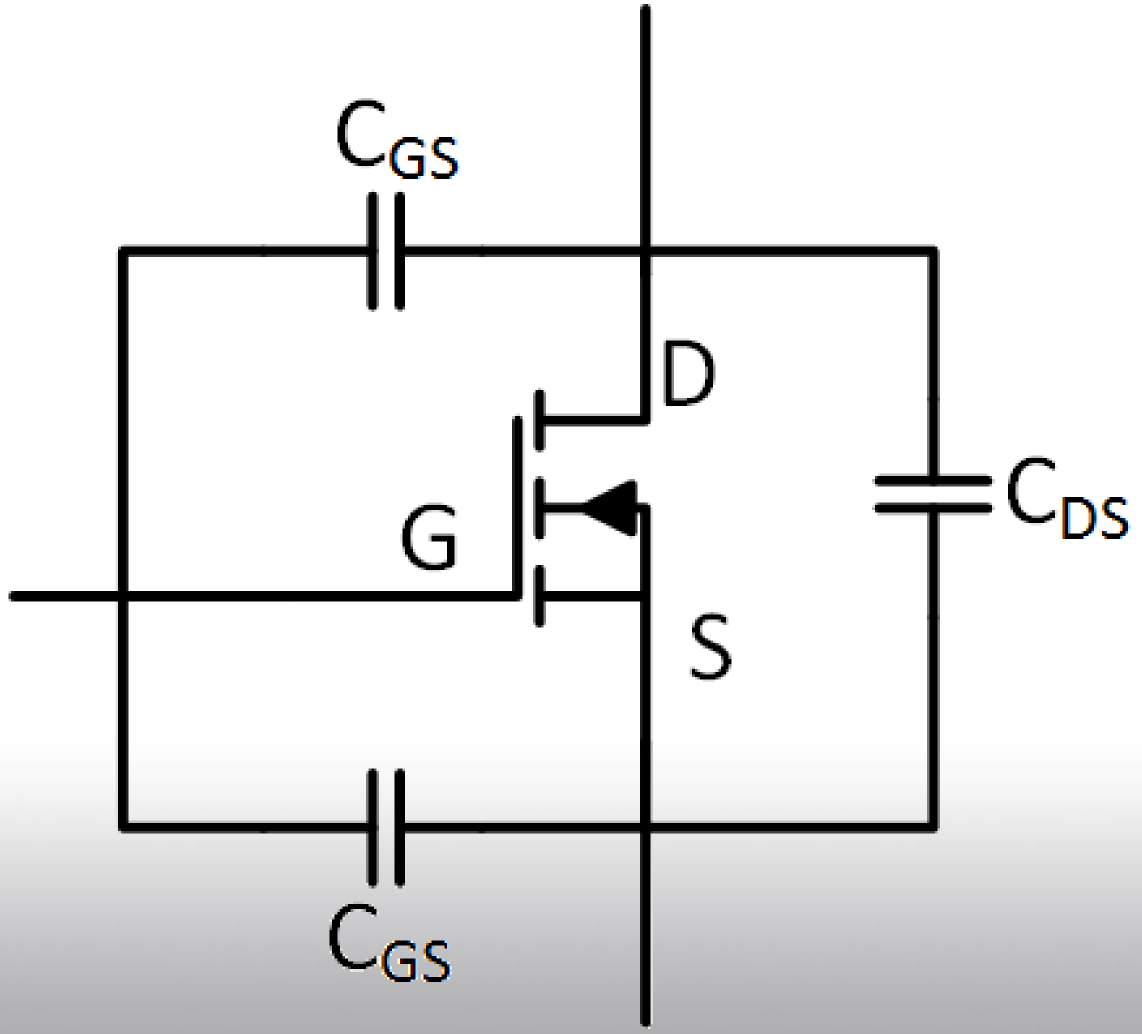
Image taken from [10].

N-Channel MOSFETs



Screenshot of oscilloscope

N-Channel MOSFETs

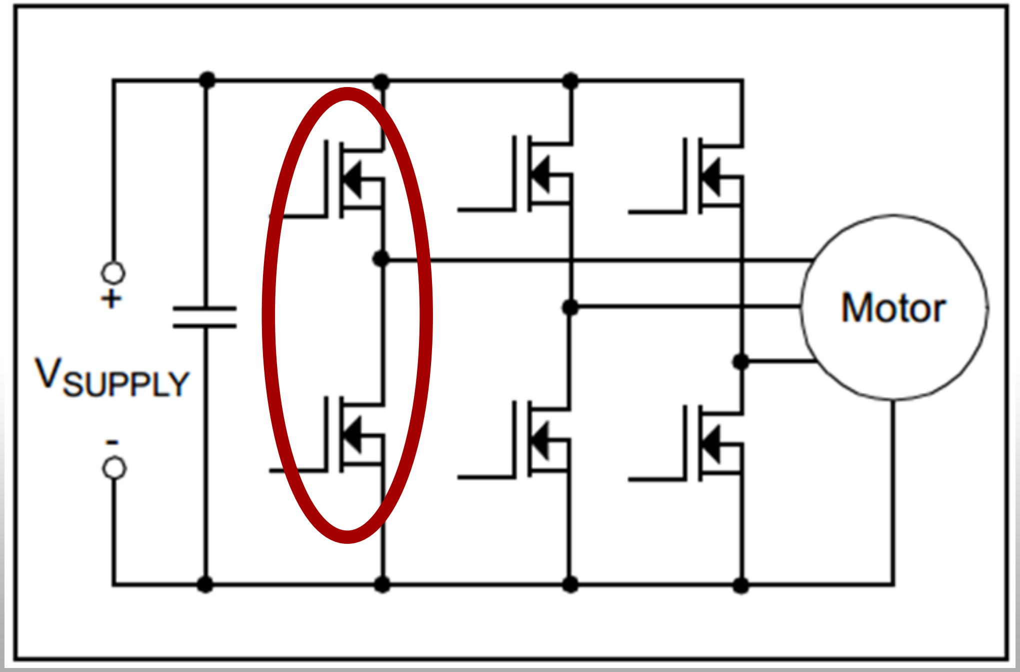


Design Considerations

- MOSFET Switching Time
  - Dead Time
  - Shoot Through

Design Considerations

- MOSFET Switching Time
  - Dead Time
  - Shoot Through



MOSFET Switching Time

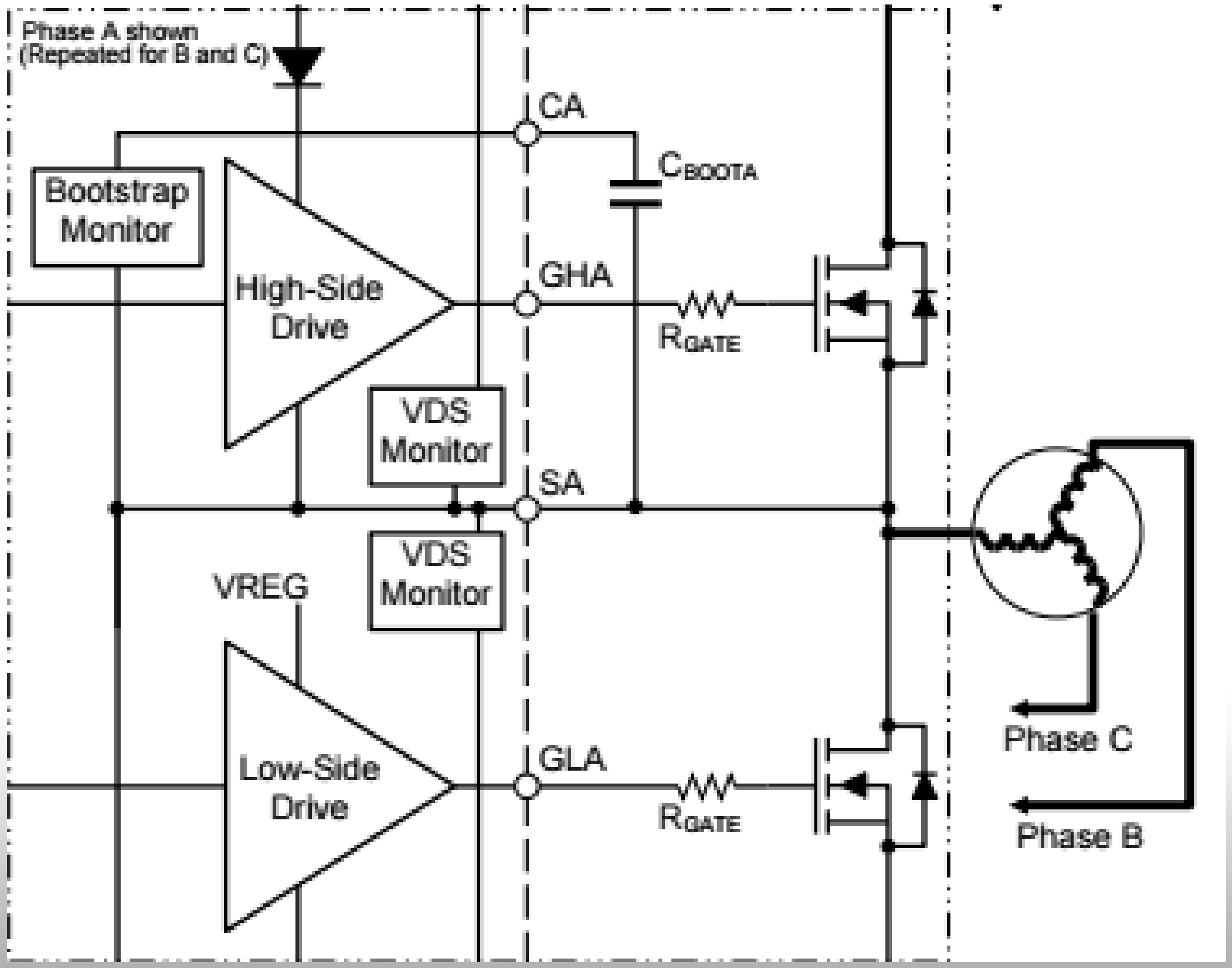
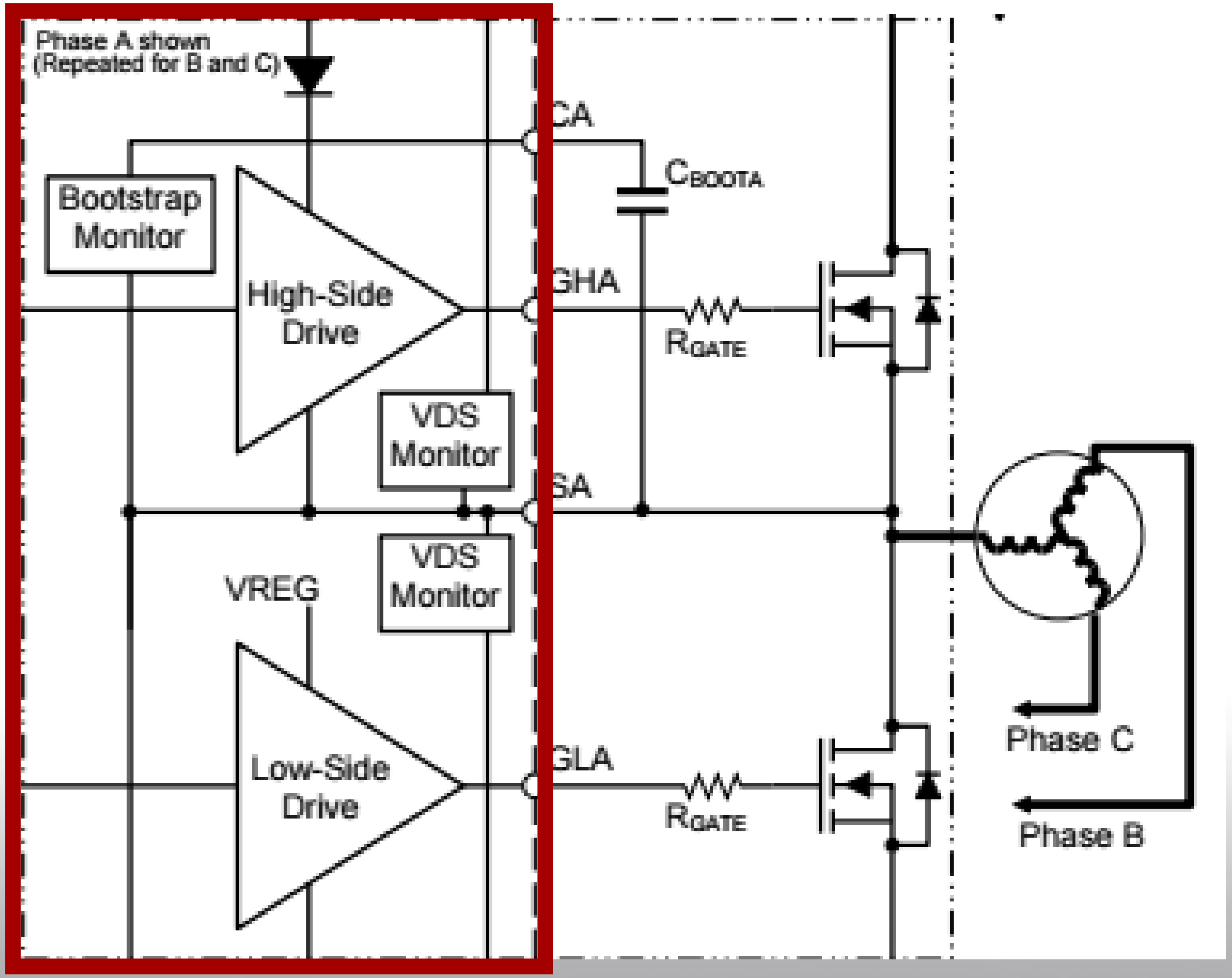
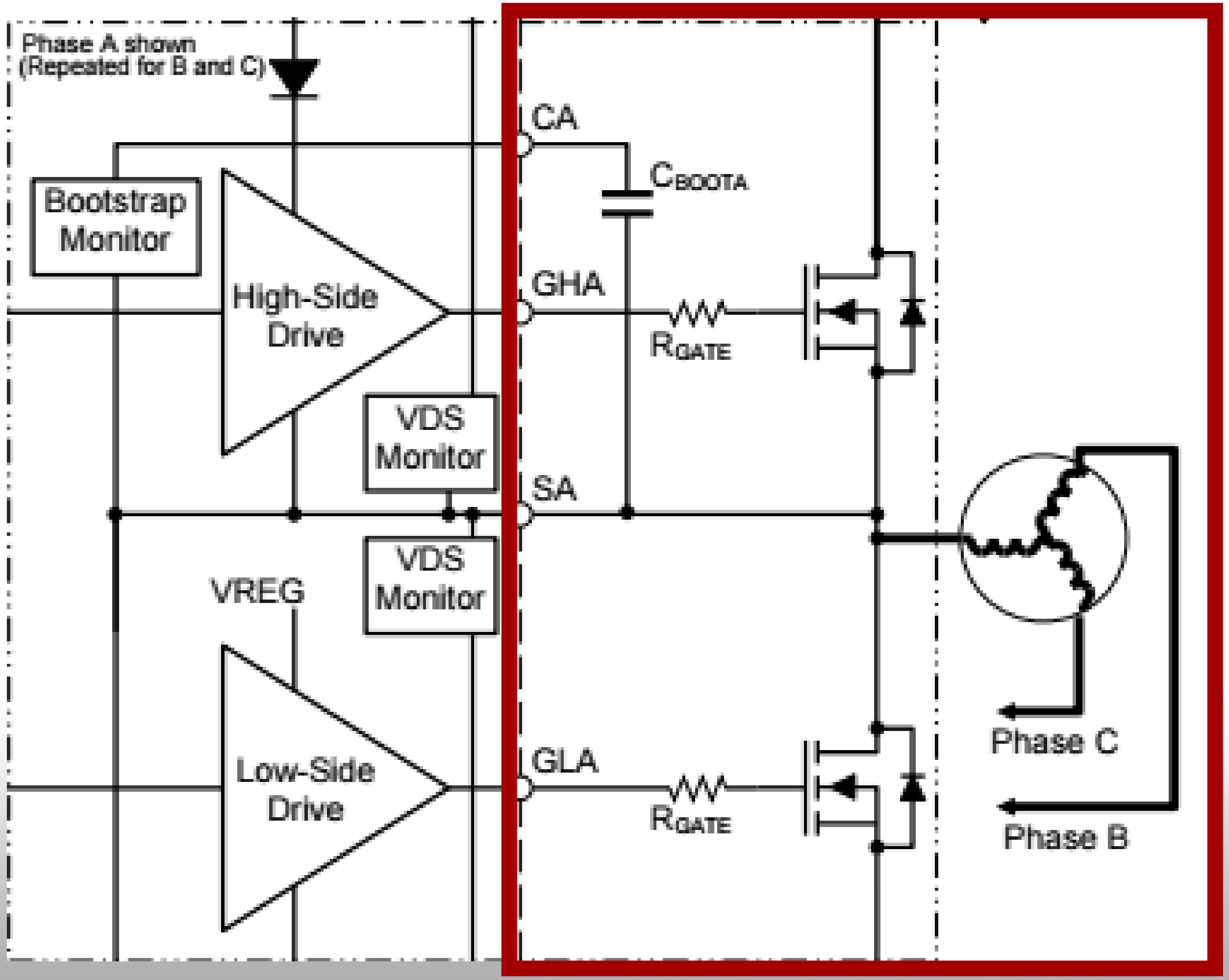


Image taken from [9].

MOSFET Switching Time



MOSFET Switching Time



MOSFET Switching Time

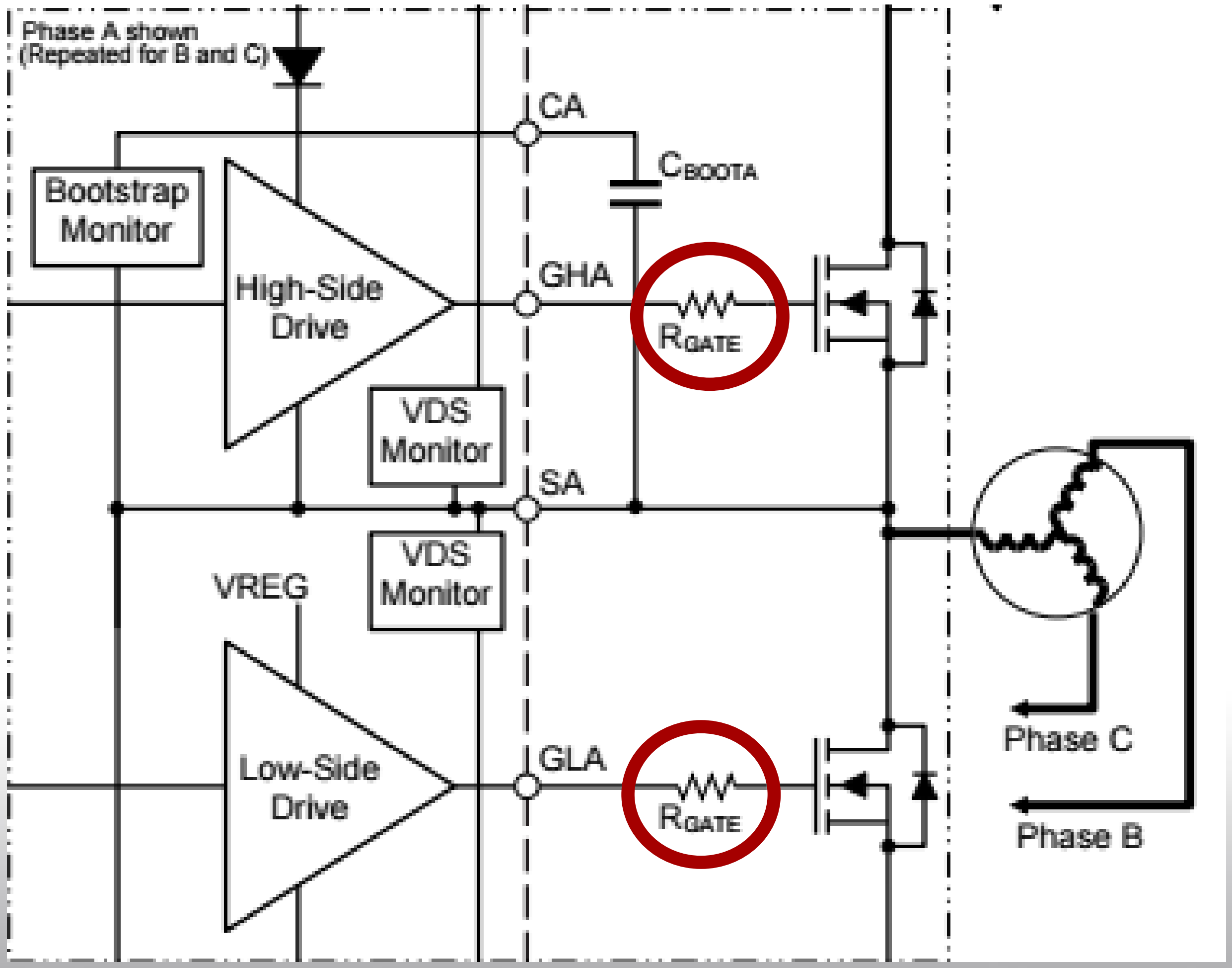


Image taken from [9].

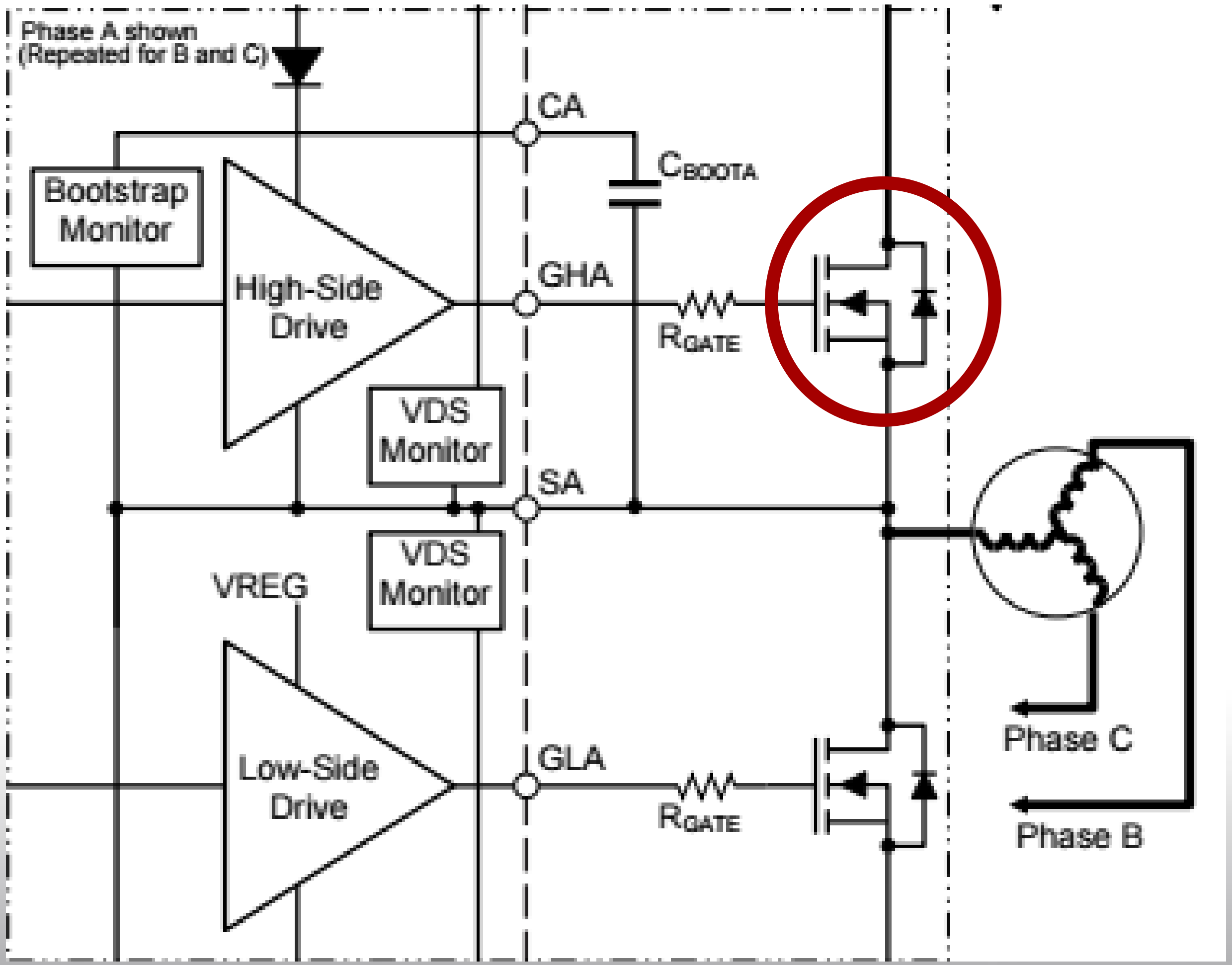
MOSFET Switching Time

- Turn On Time:
  - $R_G = \frac{t_r}{\left(C_{gs}+C_{gd}\right)\ln\left(\frac{1}{1-\left(\frac{V_{gp}}{V_{gsapp}}\right)}\right)} - R_{gapp}$
- Turn Off Time:
  - $R_G = \frac{t_f}{\left(C_{gd}+C_{gs}\right)\ln\left(\frac{V_{gsapp}}{V_{gp}}\right)} - R_{gapp}$

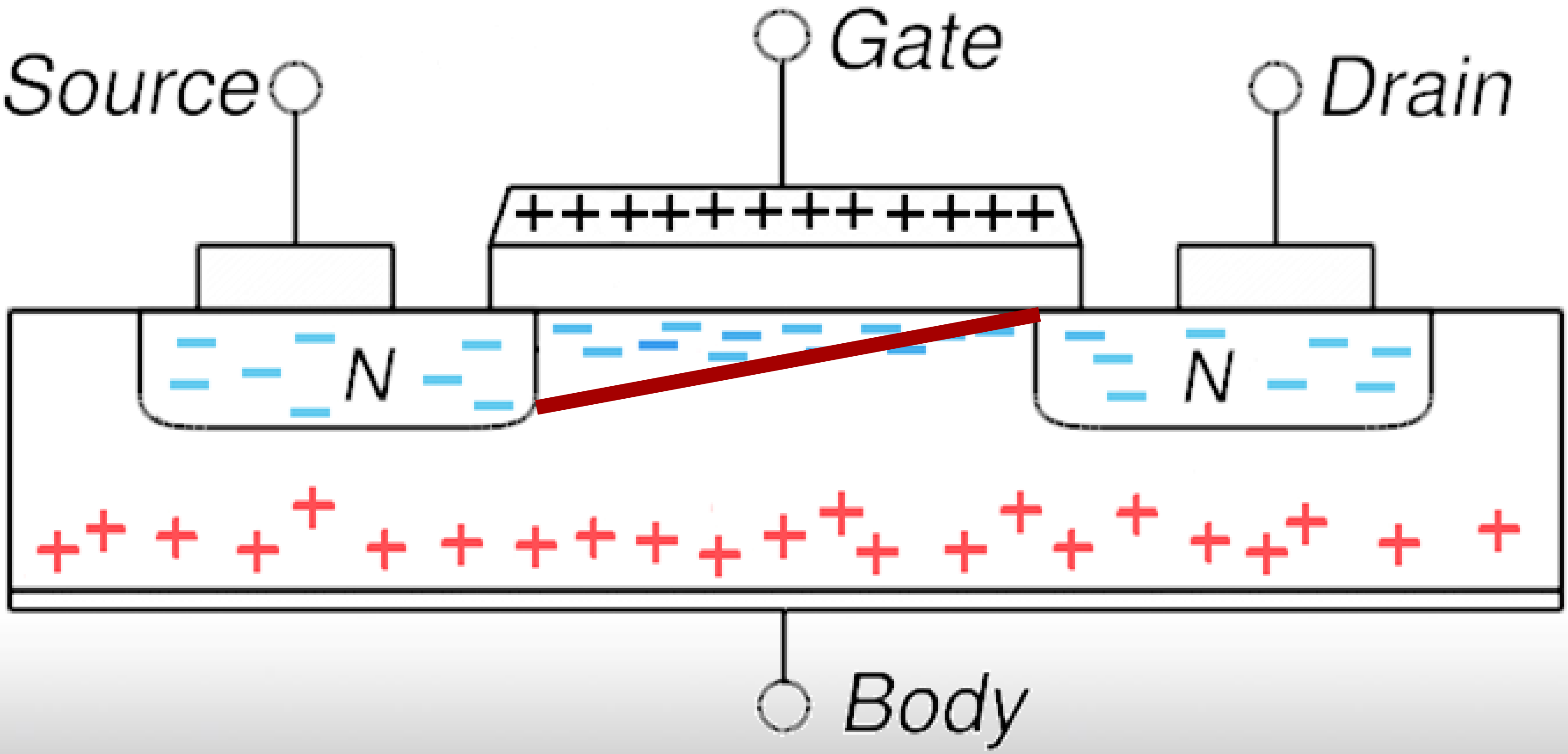
Design Considerations

- MOSFET Switching Time
- N-Channel MOSFETs
  - High Side
  - Low Side

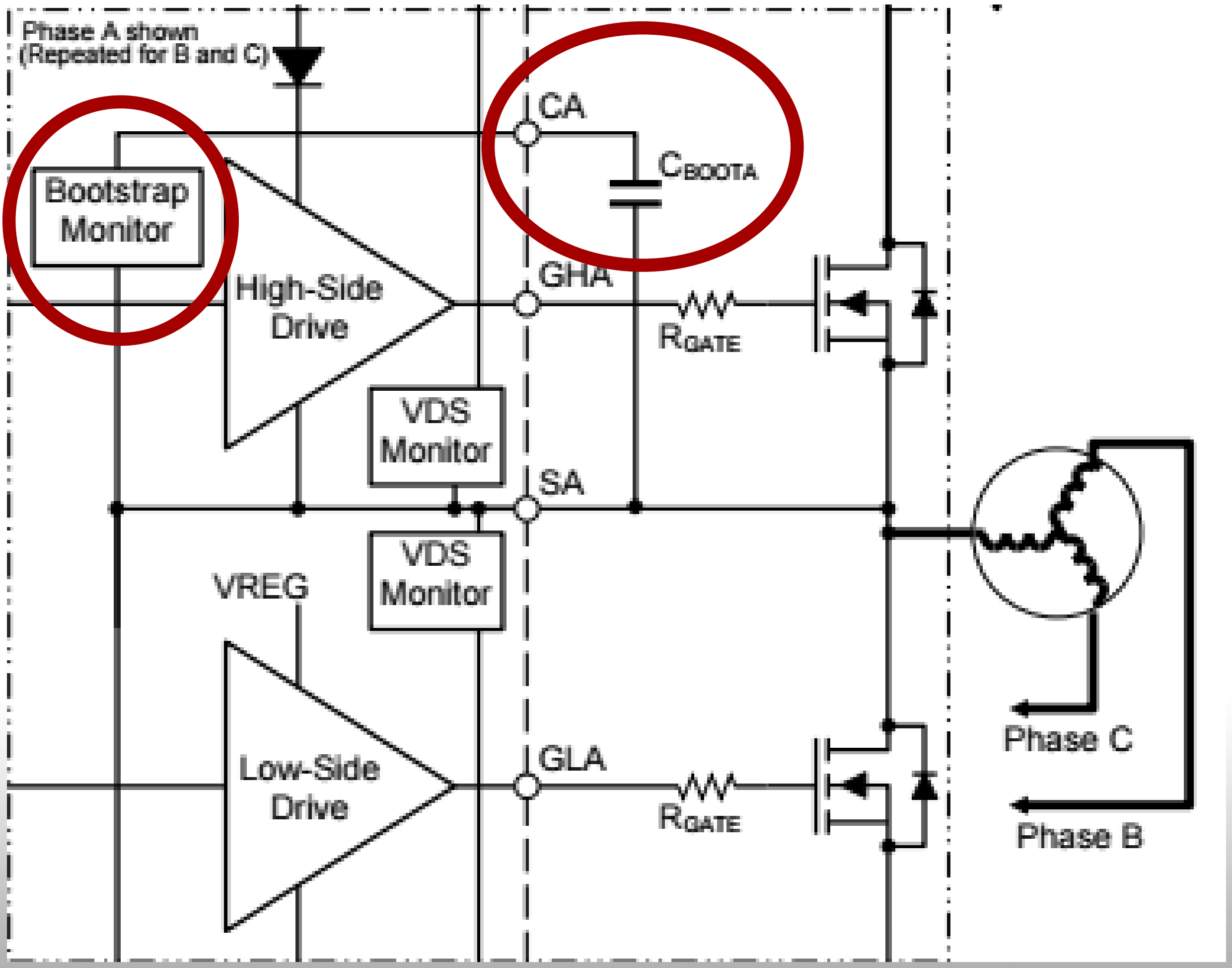
N-Channel MOSFETs – High Side



N-Channel MOSFETs – High Side



N-Channel MOSFETs – High Side



N-Channel MOSFETs – High Side

- $\Delta \mathrm{V}_{\mathrm{BOOT}} = \mathrm{V}_{\mathrm{DD}} - \mathrm{V}_{\mathrm{F}} - \mathrm{V}_{\mathrm{GSMIN}}$

- $Q_{TOTAL} = Q_{GATE} + (i_{LKGS}) * t_{ON} + Q_{LS}$

- $C_{BOOT} = \frac{Q_{TOTAL}}{\Delta \mathrm{V}_{\mathrm{BOOT}}}$

N-Channel MOSFETs – Low Side

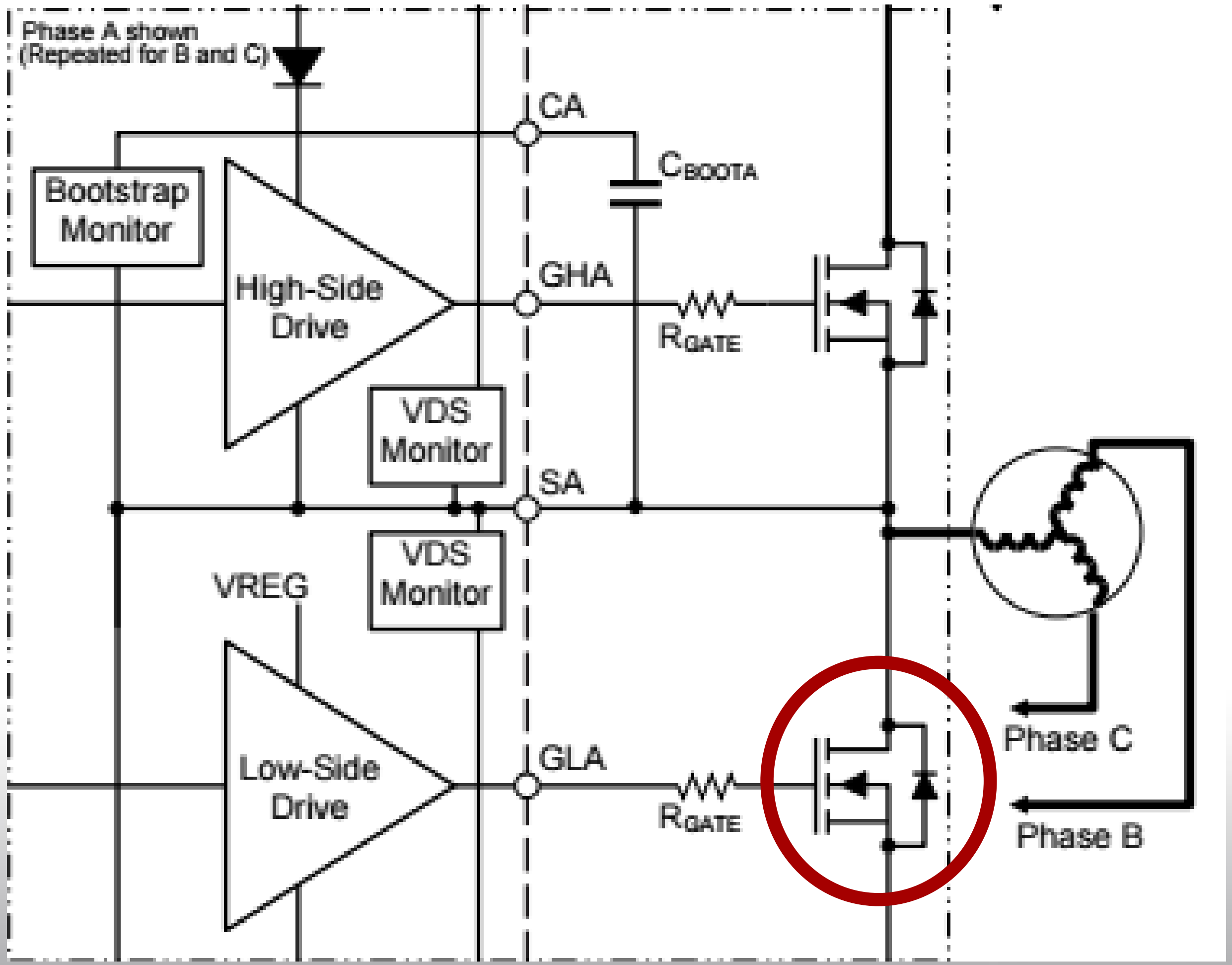


Image taken from [9].

N-Channel MOSFETs – Low Side

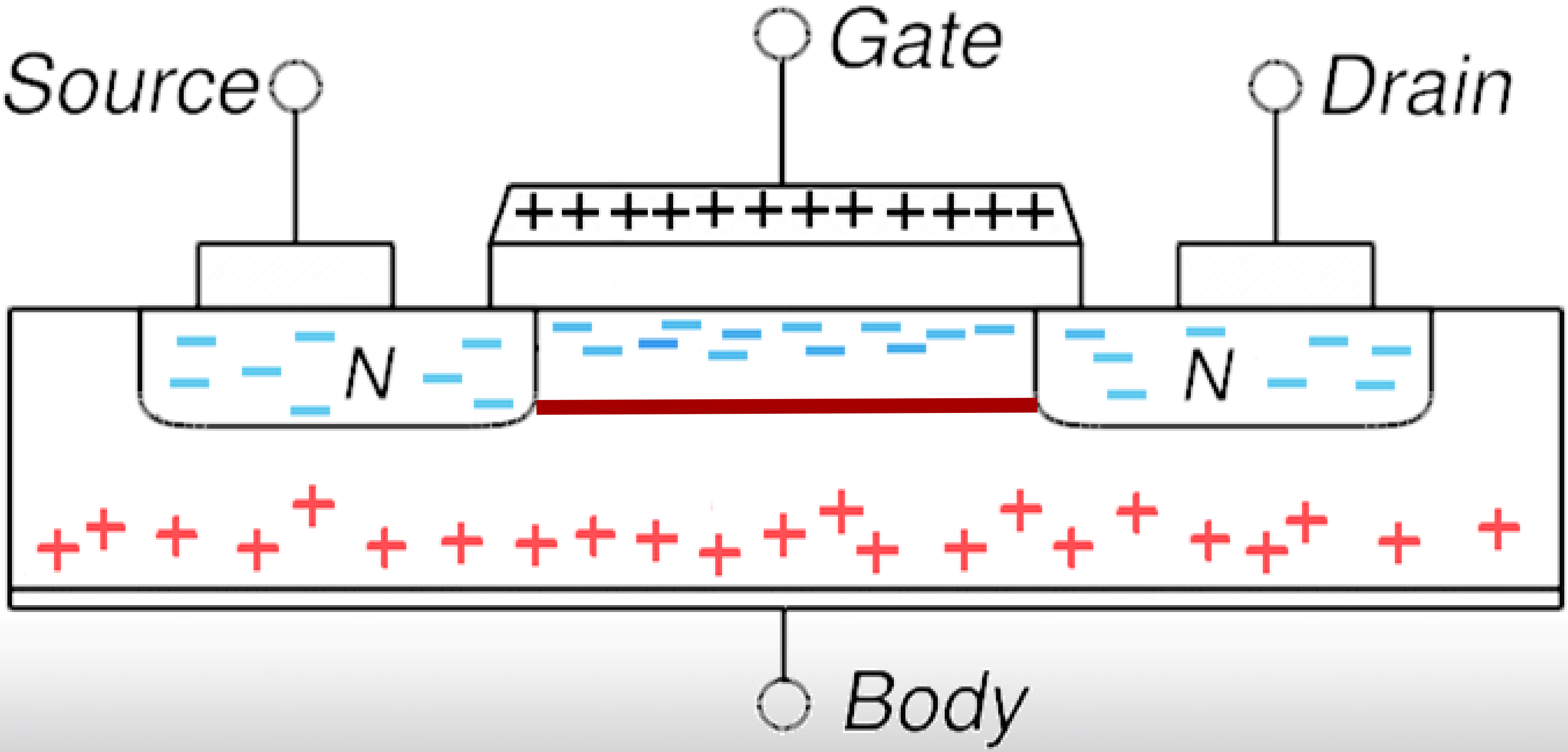
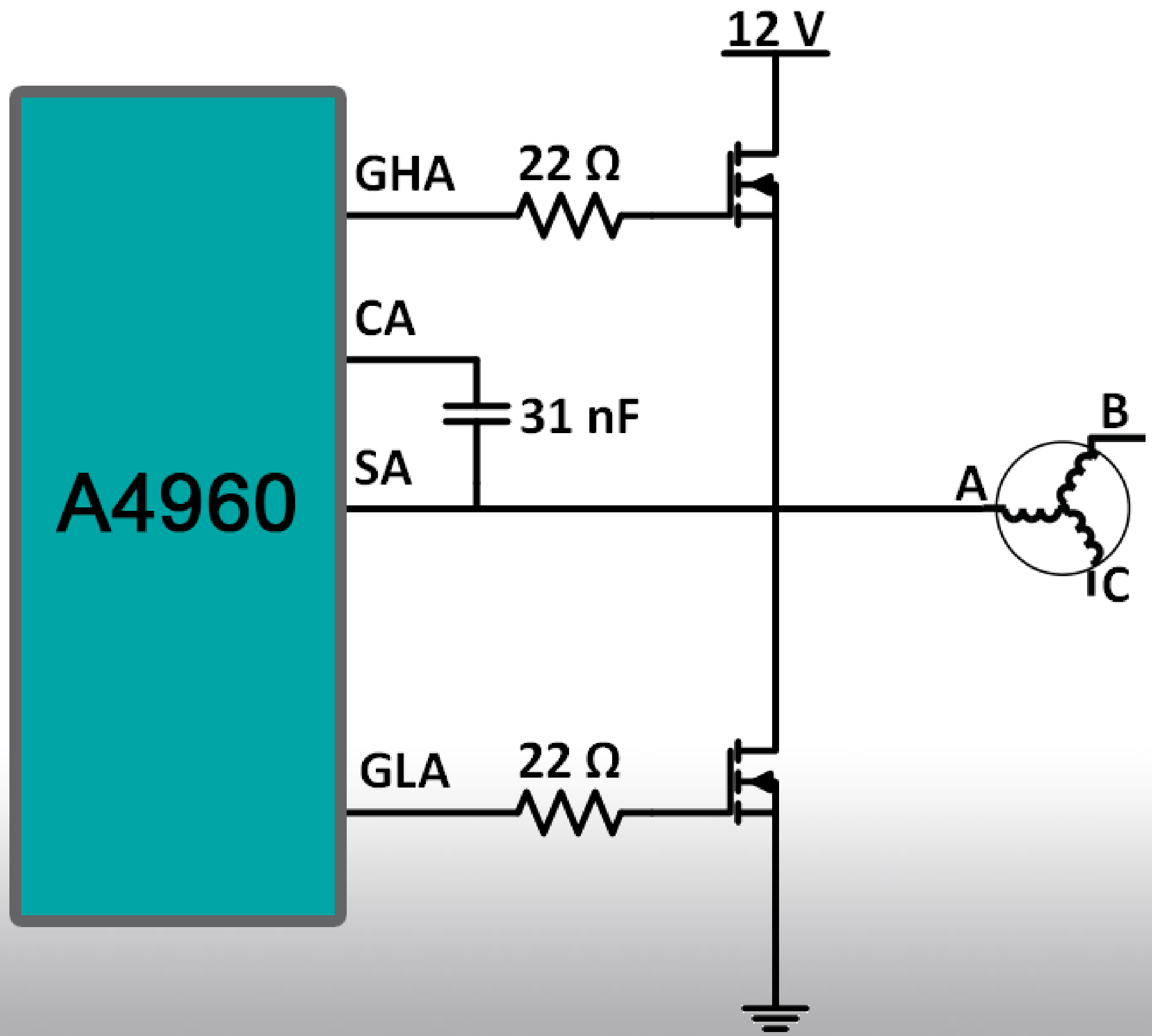


Image taken from [10].

Phase A Predriver Interface



Presentation Outline

- Background
- Evan Dinelli
- Dan Van de Water
- Michael Barnes
- Summary and Conclusions
- Questions and Answers (Q & A)

Summary And Conclusions

- Framework for RoboBoat
  - Navigation
  - Thrust

Image taken from [11].

Summary And Conclusions

- GPS and Compass Data Processing
- RC Commands
- Master-Slave Communication
- Designed Power Transistor Circuit

Presentation Outline

- Background
- Evan Dinelli
- Dan Van de Water
- Michael Barnes
- Summary and Conclusions
- Questions and Answers (Q & A)

Navigation and Thrust Systems for AUVSI RoboBoat

Team: Michael S. Barnes, Evan J. Dinelli, Daniel R. Van de Water
Advisors: Mr. Nick Schmidt and Dr. Gary Dempsey

BRADLEY University

Department of Electrical and Computer Engineering

April 26th, 2016

References

- [1] http://www.auvsifoundation.org/foundation/competitions/competition-central/roboboat
- [2] http://www.bradley.edu/inthespotlight/story/?id=b46cf284-2bd9-4efb-917e-ba6ca565cf84
- [3] http://www.bluerobotics.com/thruster/
- [3] http://www.amazon.com/Futaba-2-4Ghz-Helicopter-Aircraft-Transmitter/dp/B0015H6FOC
- [4] http://images.amain.com/images/large/fut/futl7627.jpg
- [5] http://www.electricaleasy.com/2015/01/how-does-servo-motor-work.html
- [6] http://www.allegromicro.com/~/Media/Files/Datasheets/A4960-Datasheet.ashx
- [7] https://www.youtube.com/watch?v=oFI7VW6WGR4
- [8] Jamie Dunn, “Determining MOSFET Driver Needs for Motor Drive Applications,” Microchip Technology Inc., 2003.
- [9] http://www.allegromicro.com/~/Media/Files/Datasheets/A4960-Datasheet.ashx
- [10] http://volga.eng.yale.edu/index.php/main/semiconductors
- [11] http://www.auvsifoundation.org/competitions/competition-central/roboboat/past-roboboat-competitions
- [12] https://higherlogicdownload.s3.amazonaws.com/AUVSI/fb9a8da0-2ac8-42d1-a11e-d58c1e158347/UploadedFiles/RoboBoat_2014_final_rules.pdf
- [13] http://www.gpsinformation.org/dale/nmea.htm
- [14] http://www.robot-electronics.co.uk/htm/cmps10i2c.htm

References

- [15] AUVSI Foundation. (2015). RoboBoat - Rules [Online]. Available: https://higherlogicdownload.s3.amazonaws.com/AUVSI/fb9a8da0-2ac8-42d1-a11e-d58c1e158347/UploadedFiles/RoboBoat_2015_final_rules.pdf
- [16] NEMA Data [Online]. Available: “AUVSI foundation” Available: http://www.auvsifoundation.org/home
- [17] ATmega1284 Datasheet [Online]. Available: http://www.atmel.com/images/Atmel-8272-8-bit-AVR-microcontroller-ATmega164A_PA-324A_PA-644A_PA-284_P_datasheet.pdf
- [18] “Atmega 644A Datasheet” http://www.atmel.com/Images/Atmel-8272-8-bit-AVR-microcontroller-ATmega164A_PA-324A_PA-644A_PA-1284_P_datasheet.pdf
- [19] Futaba T66EX 2.4 GHZ Datasheet [Online]. Available: http://manuals.hobbico.com/fut/6ex-2_4ghz-manual.pdf

Extra Slides

Competition Area

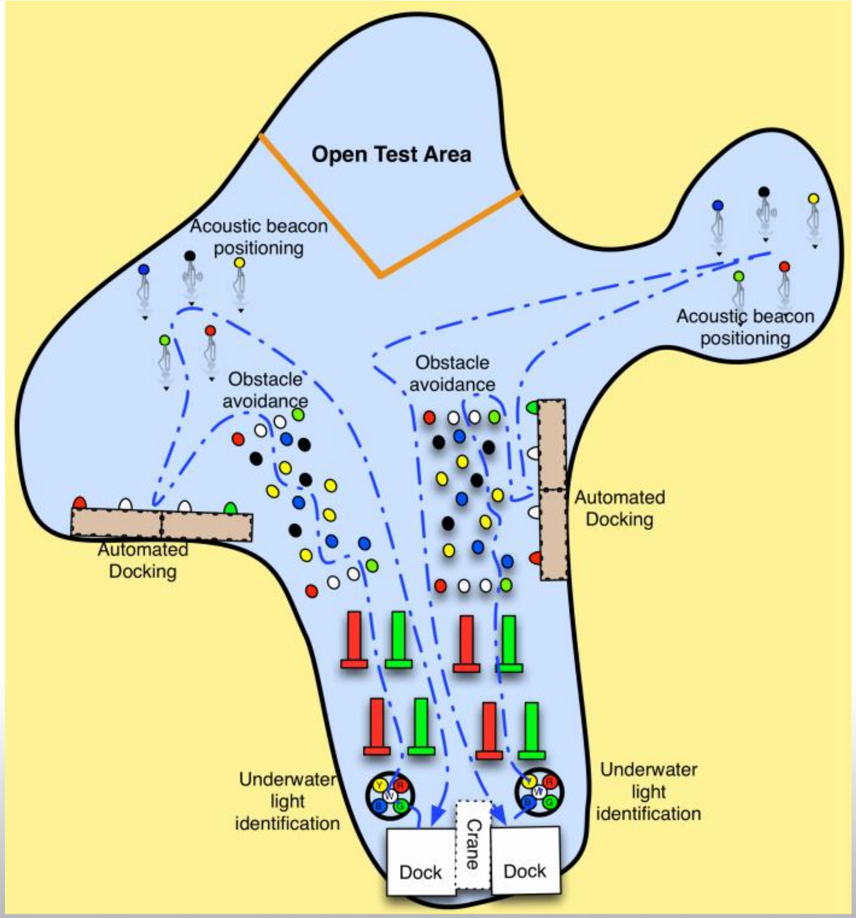


Image taken from [12].

Constraints - AUVSI

	2013	2014	2015
Communication			
Energy source			
Kill switch			
e-Kill switch			
Propulsion			
Remote control			
Safety			
Size			
Waterproof			
Weight			

Constraints and Requirements

Communication
Energy source
Kill switch
e-Kill switch
Propulsion
Remote control
Safety
Size
Waterproof
Weight
Cost
GPS and compass
Mode switch
Reusable

Constraints and Requirements

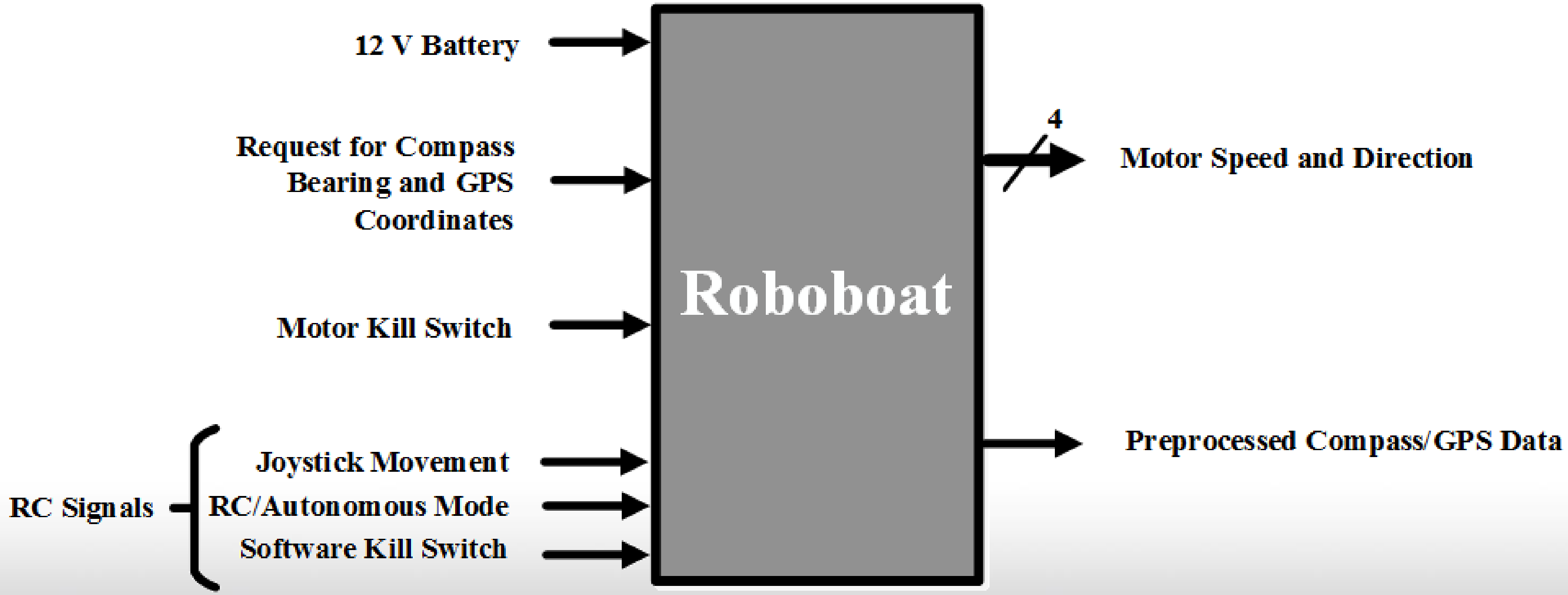
	Type
Communication	Constraint
Energy source	Constraint
Kill switch	Constraint
e-Kill switch	Functional Requirement
Propulsion	Functional Requirement
Remote control	Functional Requirement
Safety	Nonfunctional Requirement
Size	Constraint
Waterproof	Nonfunctional Requirement
Weight	Constraint
Cost	Constraint
GPS and compass	Functional Requirement
Mode switch	Functional Requirement
Reusable	Nonfunctional Requirement

Specifications

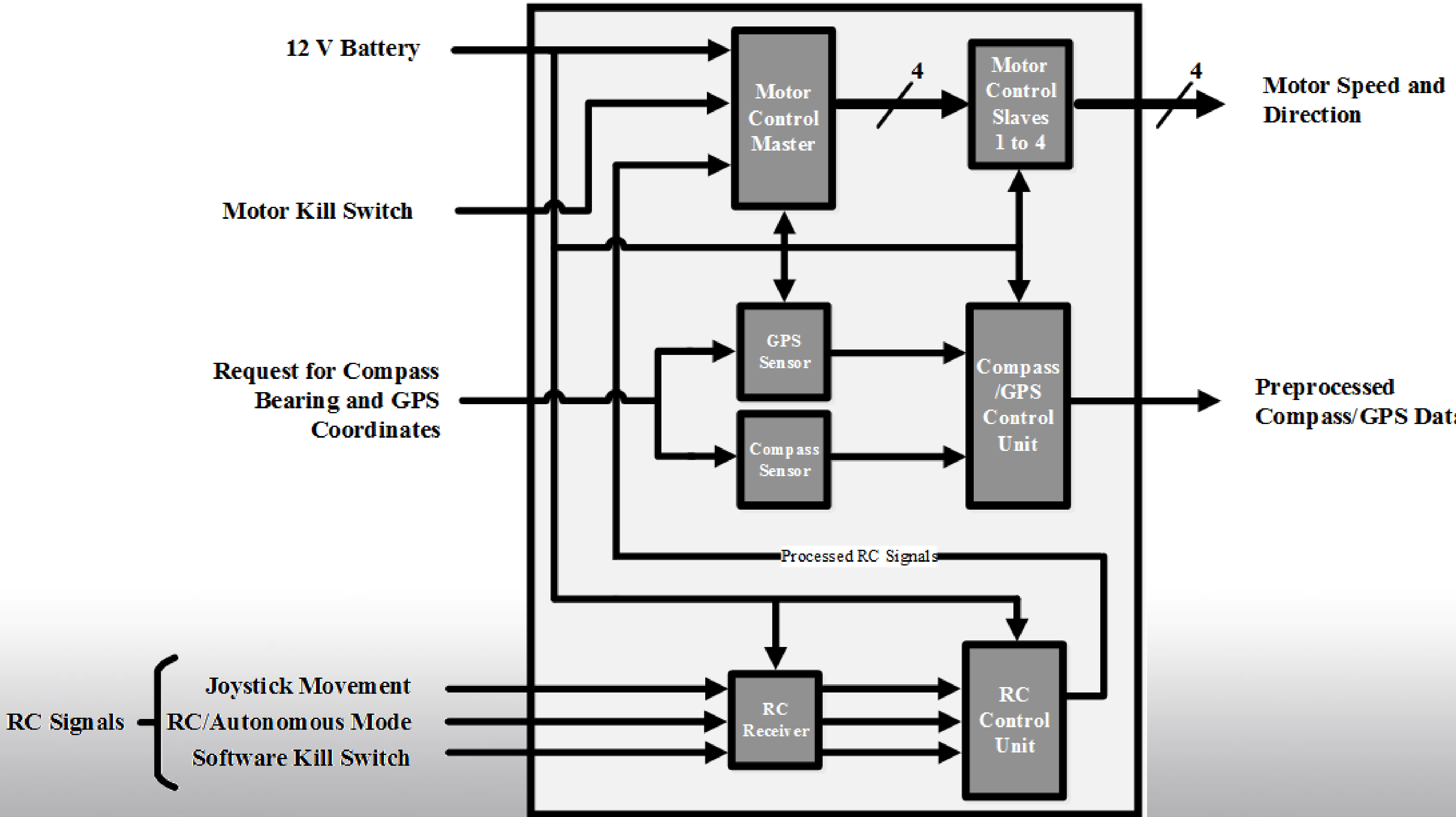
Functional Requirements	Specifications
GPS and Compass	GPS Accuracy: < ±2 m
	Compass Accuracy: < ±2°
Remote Control	Rate: ≥ 5 Commands / sec
	Mode Signal
	Software Kill Signal
Motor Control	Rate: ≥ 5 Commands / sec
	Physical Kill Switch

Nonfunctional Requirements
Reusability
Water Resistance

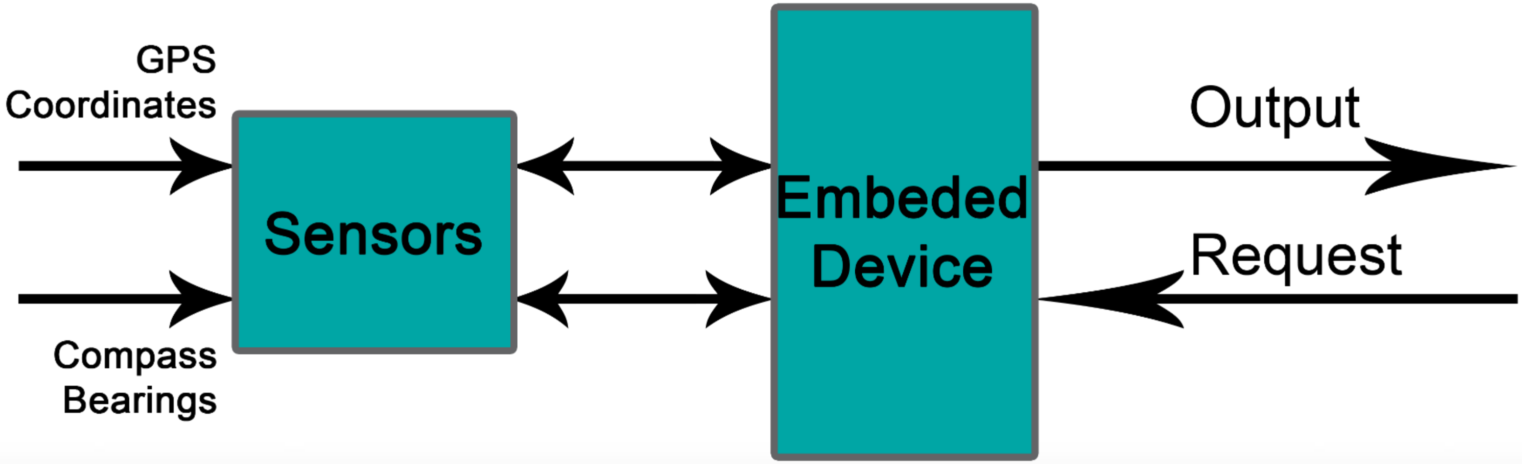
System Block Diagram



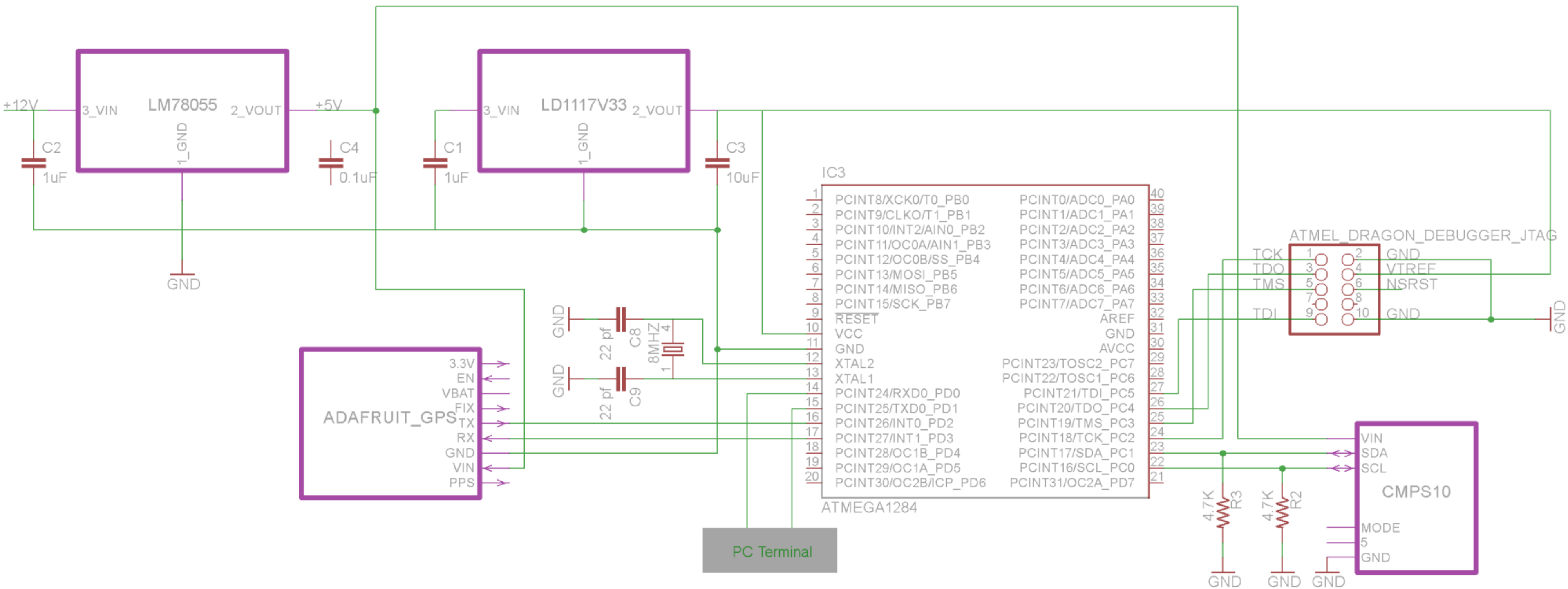
Subsystem Block Diagram



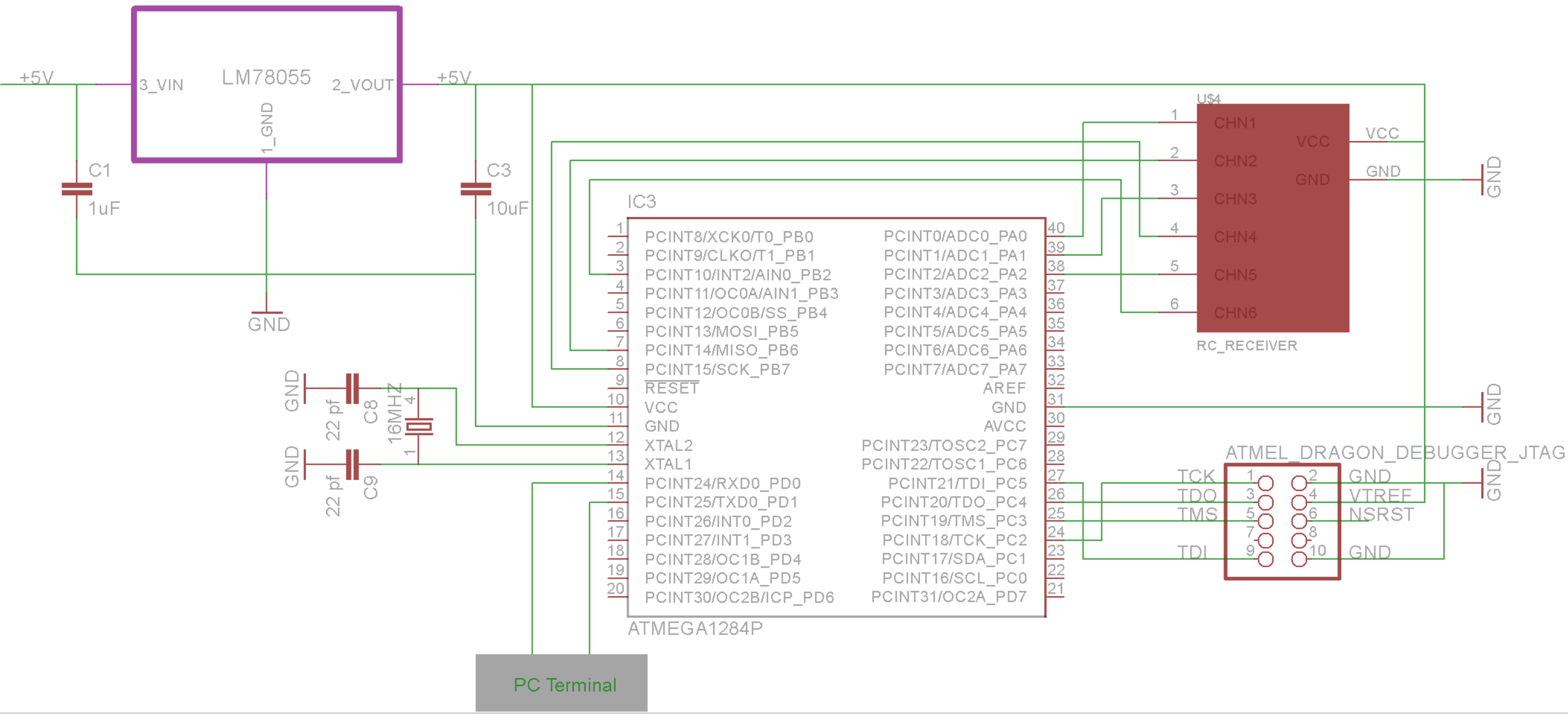
Navigation System Block Diagram



Navigation Subsystem Schematic



RC Unit Schematic



MCU and Compass Interfacing

ATmega1284		CMPS10	
SDA, receive	VIL,min = 0.99V VIH,max = 1.98-3.83V	**SDA, transmit**	3.3V levels
SDA, transmit	VOH,min = 2.3V VOL,max = 0.6V	**SDA, receive**	3.3V levels

MCU and GPS Interfacing

ATmega1284		Adafruit Ultimate GPS	
Rx	VIL,max = 0.99V	Tx	VOH,min = 2.3V
	VIH,min = 1.98V		VOL, max = 0.6V
Tx	VOH,min = 2.3V	Rx	VIH,min = 1.98V
	VOL,max = 0.6V		VIL,max = 0.99V

MCU and MAX3232 Interfacing

ATmega1284		MAX3232	
Rx	VIL,max = 0.99V	Tx	VOH,min = 2.7V
	VIH,min = 1.98V		VOL, max = 0.4V
Tx	VOH,min = 2.3V	Rx	VIH,min = 1.8-2.4V
	VOL,max = 0.6V		VIL,max = 0.6-1.2V

Baud Rate

$$Error[\%] = \left(\frac{BaudRate_{ClosestMatch}}{BaudRate} - 1\right) * 100\%$$

(1)

$$BaudRate_{ClosestMatch} = \frac{f_{osc}}{16 * BaudRate} - 1$$

(2)

USART Registers

Important Registers for USART Initialization	
UBRR1H, UBRR1L	Set Baud Rate
UCSR1B	Enable Receiver and Transmitter
UCSR1C	2 Stop Bits

Important Registers for USART Data Transmission	
UCSR1A, UDRE1	Wait for Empty Buffer
UDR1	Where to Send Data

NEMA 2.0

Name	Garmin	Magellan	Lowrance	SiRF	Notes:
GPAPB	N	Y	Y	N	Auto Pilot B
GPBOD	Y	N	N	N	bearing, origin to destination - earlier G-12's do not transmit this
GPGGA	Y	Y	Y	Y	fix data
GPGLL	Y	Y	Y	Y	Lat/Lon data - earlier G-12's do not transmit this
GPGSA	Y	Y	Y	Y	overall satellite reception data, missing on some Garmin models
GPGSV	Y	Y	Y	Y	detailed satellite data, missing on some Garmin models
GPRMB	Y	Y	Y	N	minimum recommended data when following a route
GPRMC	Y	Y	Y	Y	minimum recommended data
GPRTE	Y	U	U	N	route data, only when there is an active route. (this is sometimes bidirectional)
GPWPL	Y	Y	U	N	waypoint data, only when there is an active route (this is sometimes bidirectional)

Image taken from [13].

NEMA 2.0

GGA - essential fix data which provide 3D location and accuracy data.

```
 $GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47

Where:
     GGA          Global Positioning System Fix Data
     123519       Fix taken at 12:35:19 UTC
     4807.038,N   Latitude 48 deg 07.038' N
     01131.000,E  Longitude 11 deg 31.000' E
     1            Fix quality: 0 = invalid
                               1 = GPS fix (SPS)
                               2 = DGPS fix
                               3 = PPS fix
                               4 = Real Time Kinematic
                               5 = Float RTK
                               6 = estimated (dead reckoning) (2.3 feature)
                               7 = Manual input mode
                               8 = Simulation mode
     08           Number of satellites being tracked
     0.9          Horizontal dilution of position
     545.4,M      Altitude, Meters, above mean sea level
     46.9,M       Height of geoid (mean sea level) above WGS84
                      ellipsoid
     (empty field) time in seconds since last DGPS update
     (empty field) DGPS station ID number
     *47          the checksum data, always begins with *
```

Image taken from [13].

Raw GPS Output

- GPS Readings
  - Asynchronous Serial
  - 9600 Baud Rate, 1 Hz Transmission
  - No Parity, 1 Stop Bit

```
$GPGGA,204555.000,4041.9645,N,08936.9937,W,2,8,0.98,195.3,M,-34.0,M,0000,0000*6C
$GPGSA,A,3,14,16,26,22,23,03,29,31,,,,,1.86,0.98,1.58*0D
$GPGSV,3,1,09,26,68,132,44,03,52,269,33,16,46,180,42,31,46,052,28*76
$GPGSV,3,2,09,51,39,206,29,23,30,306,21,22,23,179,40,14,14,113,25*7A
$GPGSV,3,3,09,29,12,060,27*4B
$GPRMC,204555.000,A,4041.9645,N,08936.9937,W,0.02,296.36,160216,,,D*7D
$GPVTG,296.36,T,,M,0.02,N,0.04,K,D*36
$GPGGA,204556.000,4041.9646,N,08936.9937,W,2,8,0.98,195.2,M,-34.0,M,0000,0000*6D
$GPGSA,A,3,14,16,26,22,23,03,29,31,,,,,1.86,0.98,1.58*0D
$GPRMC,204556.000,A,4041.9646,N,08936.9937,W,0.02,296.36,160216,,,D*7D
$GPVTG,296.36,T,,M,0.02,N,0.03,K,D*31
```

Serial Stream Directly From GPS Sensor

Parsed NMEA Sentence

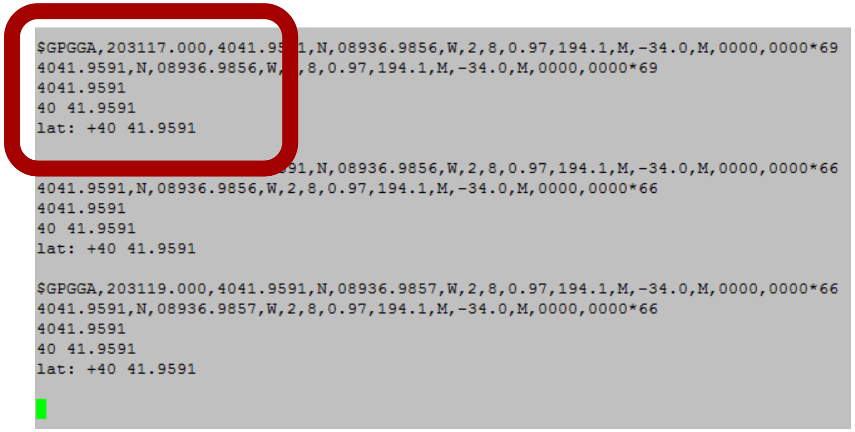
- Algorithm:
  - Find Second Comma
  - Parse:
    - Latitude
      - Degrees
      - Minutes and Seconds
  - Format
  - Repeat for Longitude

```
$GPGGA,203117.000,4041.9591,N,08936.9856,W,2,8,0.97,194.1,M,-34.0,M,0000,0000*69
4041.9591,N,08936.9856,W,2,8,0.97,194.1,M,-34.0,M,0000,0000*69
4041.9591
40 41.9591
lat: +40 41.9591

$GPGGA,203118.000,4041.9591,N,08936.9856,W,2,8,0.97,194.1,M,-34.0,M,0000,0000*66
4041.9591,N,08936.9856,W,2,8,0.97,194.1,M,-34.0,M,0000,0000*66
4041.9591
40 41.9591
lat: +40 41.9591

$GPGGA,203119.000,4041.9591,N,08936.9857,W,2,8,0.97,194.1,M,-34.0,M,0000,0000*66
4041.9591,N,08936.9857,W,2,8,0.97,194.1,M,-34.0,M,0000,0000*66
4041.9591
40 41.9591
lat: +40 41.9591
```

Serial Stream Directly From GPS Sensor

Parsed NMEA Sentence

- Algorithm:
  - Find Second Comma
  - Parse:
    - Latitude
      - Degrees
      - Minutes and Seconds
  - Format
  - Repeat for Longitude



Serial Stream Directly From GPS Sensor

Parsed NMEA Sentence

```
$GPGGA,203117.000,4041.95
4041.9591,N,08936.9856,W,
4041.9591
40 41.9591
lat: +40 41.9591
```

Serial Stream Directly From GPS Sensor

Final GPS Results

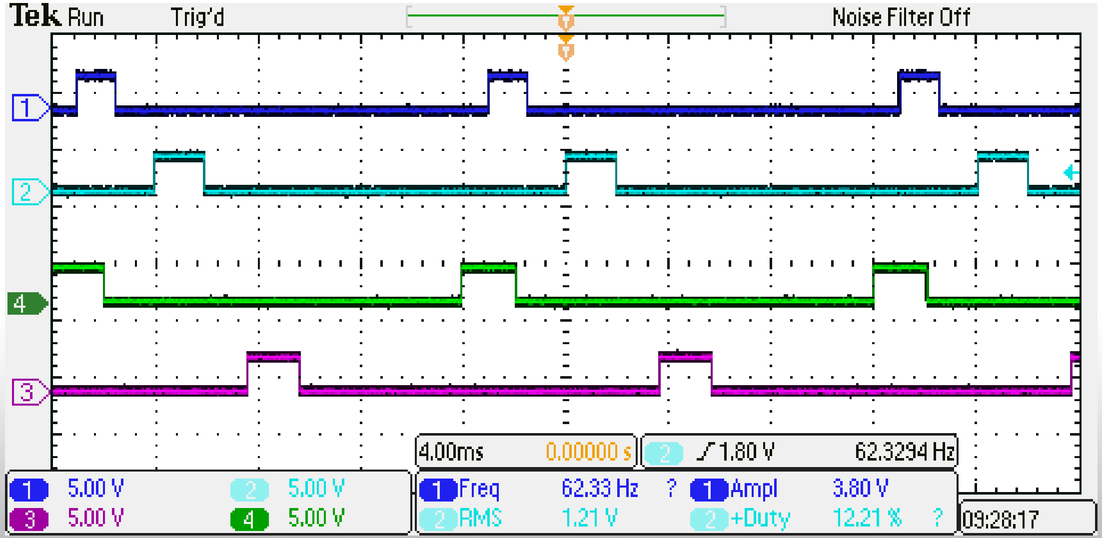
- Data Packet:
  - Latitude
  - Longitude
  - Compass Bearing (future work)

```
Position: +40 41.9624, -089 36.9951

Position: +40 41.9624, -089 36.9951

Position: +40 41.9624, -089 36.9951

Position: +40 41.9624, -089 36.9951

Position: +40 41.9623, -089 36.9951

Position: +40 41.9623, -089 36.9951

Position: +40 41.9623, -089 36.9950

Position: +40 41.9623, -089 36.9949

Position: +40 41.9623, -089 36.9949

Position: +40 41.9623, -089 36.9949
```

Serial Stream Directly From GPS Sensor

RC Design Approach

- Measure Pulse-Widths
  - Timer1, INT2, PCINT1:2, PCINT
- Mode and Kill Switches
- Convert to Motor Command



RC Receiver Output, Channels 1, 3, 5, & 6

CMPS10 Compass Registers

Register	Function
0	Software version
1	Compass Bearing as a byte, i.e. 0-255 for a full circle
2,3	Compass Bearing as a word, i.e. 0-3599 for a full circle, representing 0-359.9 degrees.
4	Pitch angle - signed byte giving angle in degrees from the horizontal plane
5	Roll angle - signed byte giving angle in degrees from the horizontal plane
6	Unused
7	Unused
8	Unused
9	Unused
10,11	Magnetometer X axis raw output, 16 bit signed integer with register 10 being the upper 8 bits
12,13	Magnetometer Y axis raw output, 16 bit signed integer with register 12 being the upper 8 bits
14,15	Magnetometer Z axis raw output, 16 bit signed integer with register 14 being the upper 8 bits
16,17	Accelerometer X axis raw output, 16 bit signed integer with register 16 being the upper 8 bits
18,19	Accelerometer Y axis raw output, 16 bit signed integer with register 18 being the upper 8 bits
20,21	Accelerometer Z axis raw output, 16 bit signed integer with register 20 being the upper 8 bits
22	Command register

Image taken from [14].

SCL Frequency

- $SCL\ freuency = \frac{CPU\ Clock\ Freqquency}{16+2(TWBR)*4^{TWPS}}$

- SCL Frequency (F_SCL) = 100 KHz
- In Software:
  - #define TWBR_val ((((F_CPU / F_SCL) / Prescaler) - 16) / 2)

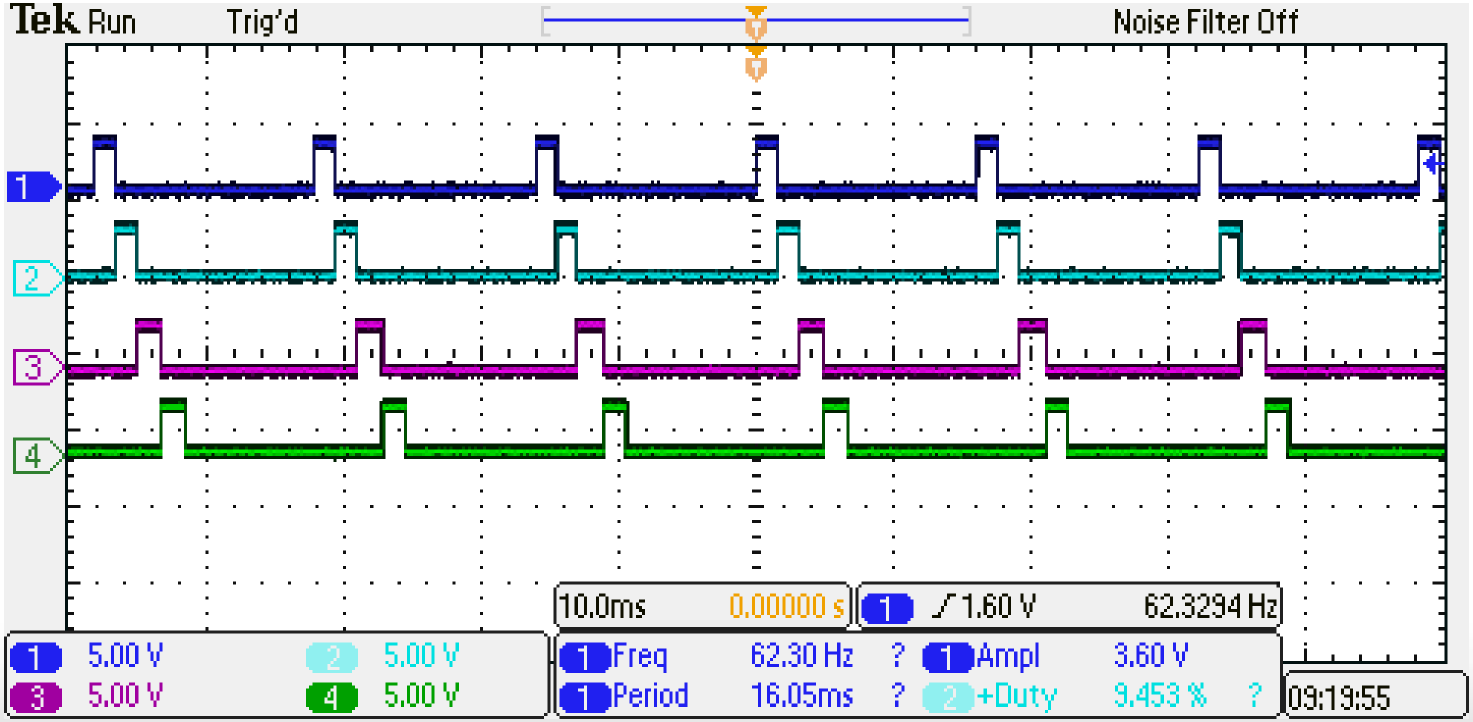
I2C Write Command

- Send Start Bit
- Load Address of I2C Device and Transmit
- Load Register Number and Transmit
- Load Write Data and Transmit
- Stop Bit

I2C Read Command

- Send Start Bi
- Load Address of I2C Device
- Transmit Data
- Send Repeated Start Bit
- Transmit Address of I2C Device
- Set Read Bit (w/ Odd Address)
- Clear Transmit Interrupt Flag
- Transmit Nack (Last Byte Request
- Read the Target Data
- Send Stop Bit on I2C Bus

RC Channels



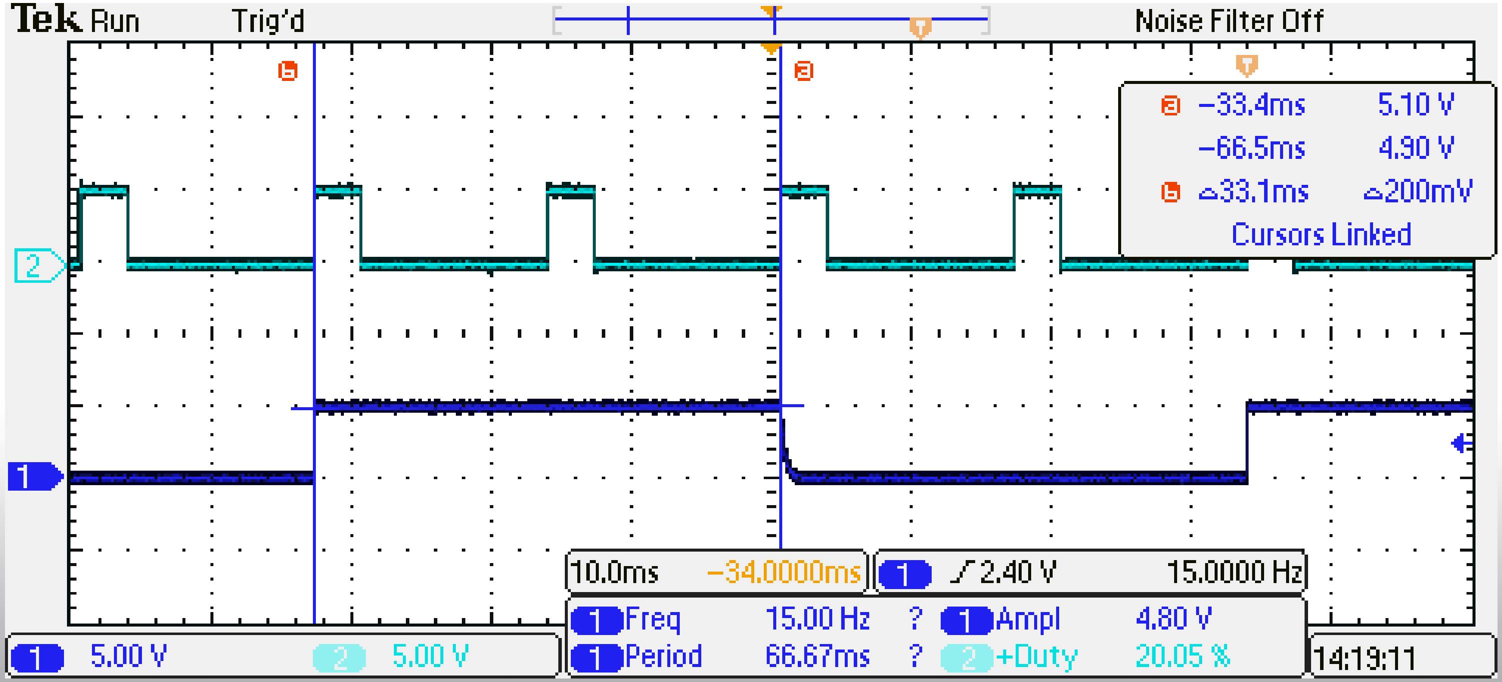
Screenshot of oscilloscope

RC Channels

RC Channel	Interrupt	ISR Action
1	PCINTA	Chk = 0 on Rising-edge
2	PCINTB	Chk = 1 on Rising-edge
3	PCINTA	Chk = 2 on Rising-edge
4	PCINTB	Chk = 3 on Rising-edge
5	PCINTA	Chk = 4 on Rising-edge and Chk = 5 on Falling-edge
6	INT2	Calculate Value of Pulse-width(Within ISR)

RC Timing

- Oscilloscope Channel 1 Toggled at the Start of RC Data Transmission
- 66.5 ms to Transmit an RC Motor Command

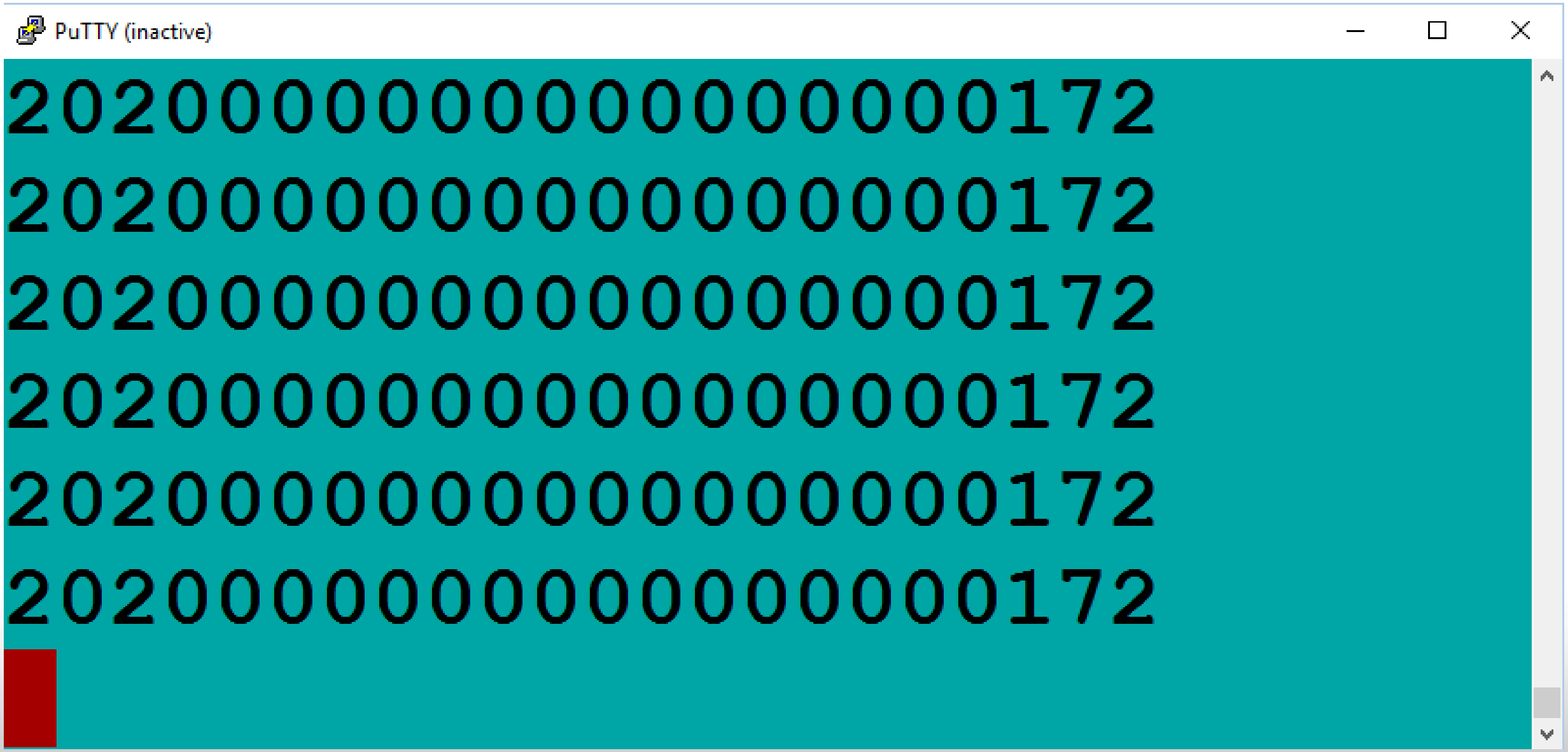


Screenshot of oscilloscope

RC Motor Command Scheme

- Neutral “Dead” Zone
- Header
- Data Bits
- Footer
- 8-bits Total

Kill Motors Command



Autonomous Mode Command

PuTTY (inactive)

```
2020110010000000000172
2020110010000000000172
2020110010000000000172
2020110010000000000172
2020110010000000000172
2020110010000000000172
```

RC Motor Commands

PuTTY (inactive)

```
202091172
20210091172
202091172
20210091172
202091172
20210091172
```

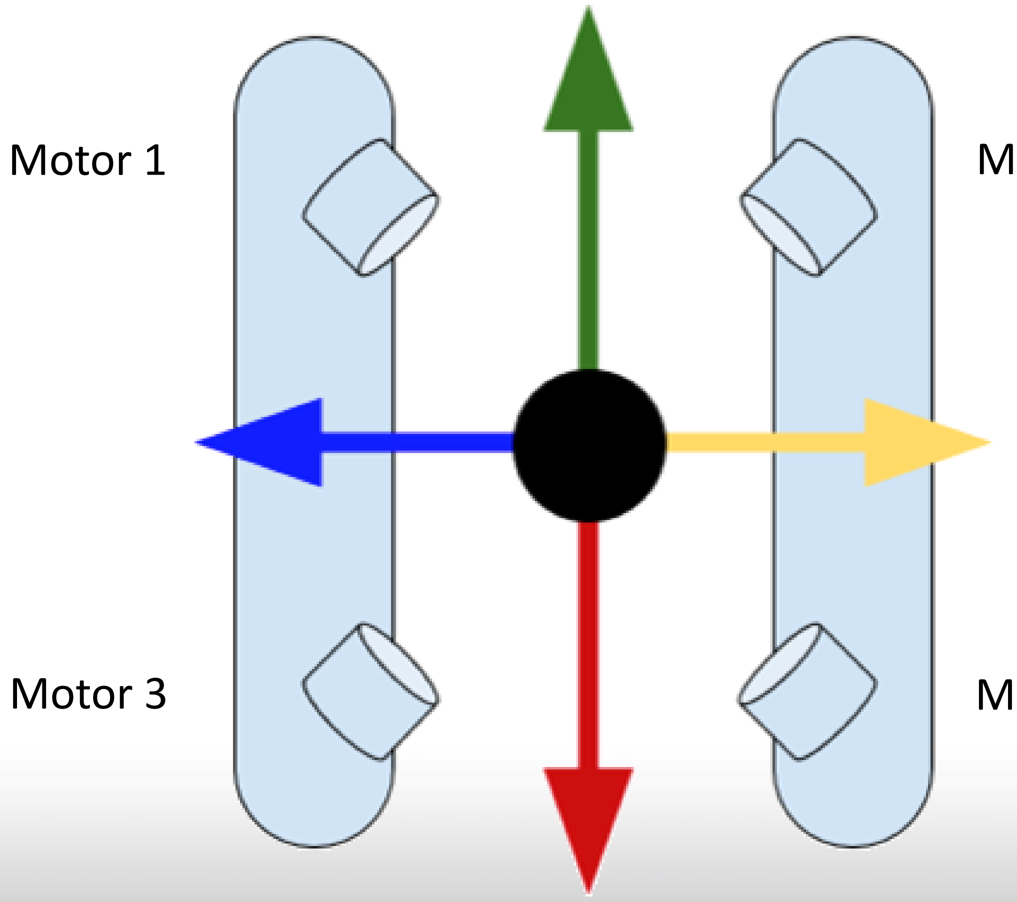
PWM Control

Forward

Backward

Left

Right



SPI Communication

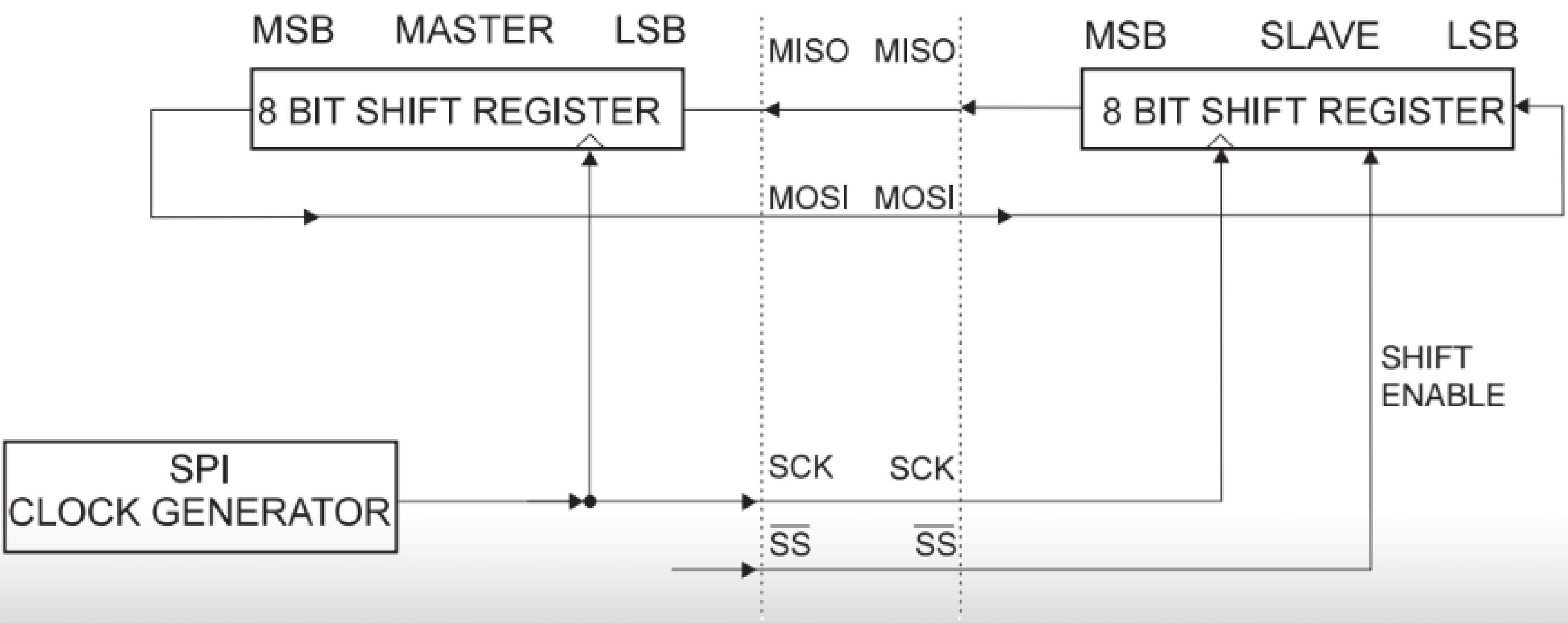
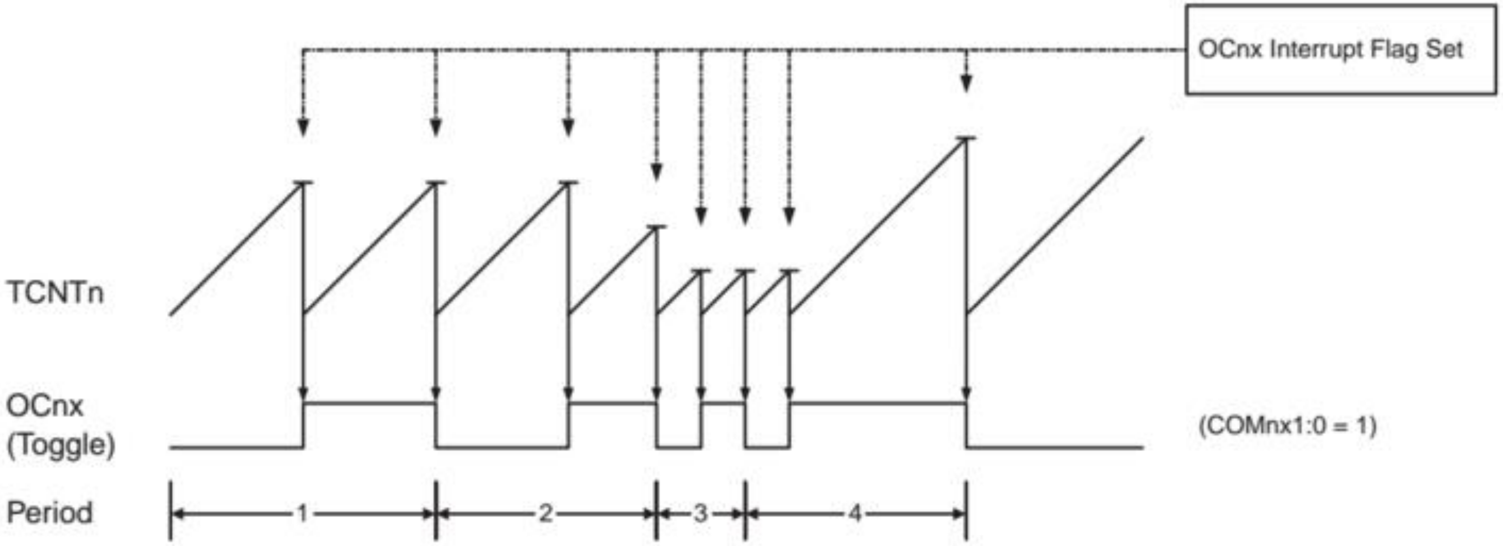


Figure taken from [15]

Interrupt Driven Code

- PWM

Figure 17-5. CTC mode, timing diagram.



Watchdog Timer

```
ISR(TIMER2_COMPA_vect){
    OCR2A = pwm2;
    __asm__("wdr;");
}
```

Watchdog Timer Information

Table 9-2. Watchdog Timer Prescale Select

WDP3	WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at $V_{CC} = 5.0V$
0	0	0	0	2K (2048) cycles	16 ms
0	0	0	1	4K (4096) cycles	32 ms
0	0	1	0	8K (8192) cycles	64 ms
0	0	1	1	16K (16384) cycles	0.125s
0	1	0	0	32K (32768) cycles	0.25s
0	1	0	1	64K (65536) cycles	0.5s
0	1	1	0	128K (131072) cycles	1.0s
0	1	1	1	256K (262144) cycles	2.0s

Motor Control Current Draw

- A4960 TOTAL: 30.253
  - VDD 16 mA
  - VBB 14 mA
  - Reference Current 3 uA
  - VBRG 250 uA
- LM7805 8 mA
- Atmega1284P 7.5 mA TOTAL: 8.2502 mA
  - PRUSART0 88.5 uA
  - PRTW1 230.3 uA
  - PRTIM3 105.5 uA
  - PRTIM1 113.7 uA
  - PRSPI 212.2 uA
- Atmega644A 9 mA TOTAL: 10.165 mA
  - PRTW1 315 uA
  - PRTIM3 300 uA
  - PRTIM1 150 uA
  - PRSPI 400 uA

TOTAL: 121.012 + 8 + 8.2502 + 40.66 = 177.92222 mA = **178 mA**

Power Loss Calculation

- Conducting Loss:
  - $P_C = R_{DSon} * (i_D)^2$
- Switching Loss:
  - $P_s = \frac{C_{rss} * (V_{in})^2 * f_{sw} * I_{LOAD}}{I_{GATE}}$

Battery

- Lithium Iron Phosphate
- 15 Amp Hour Battery (x2)
- Max Current Output Limited to 25 A
- Batteries Turn Off at 11 V
- Max Voltage of 13 V

Detailed Budget

Part	Unit Cost	Quantity	Total Cost
Blue Robotics T100 Thrusters	$109.00	4	$436.00
Internal Rectifier IRLB8748PbF HEXFET	$0.72	24	$17.28
Master Controller - ATmega 1284	$7.67	1	$7.67
Slave Controller - ATmega 644	$6.75	4	$27.00
Allegro MicroSystems A4960	$7.01	4	$28.04
Futaba T6EX Transmitter	$150.00	1	$150.00
Futaba 617FS Reciever	$69.98	1	$69.98
RC Controller - ATmega 328	$3.24	1	$3.24
Adafruit Ultimate GPS Breakout	$39.95	1	$39.95
Compass - CMPS 10	$57.33	1	$57.33
GPS/Compass Controller – Atmega 1284	$7.67	1	$7.67
			$844.16

Reusability

- Future Work
  - GPS and Compass Unit Integration
  - Fix Predriver Errors
  - Integrate MOSFET and Motor Control
  - Construct
  - Water Resistance