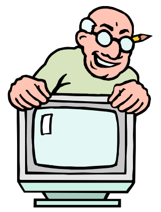
QuadAuto MKII (ESP32)

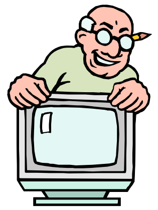
Project Overview Slides



Slides that aim to explain key features of this project.

Issue: 1.4 Released: 14/02/2026 TechKnowTone

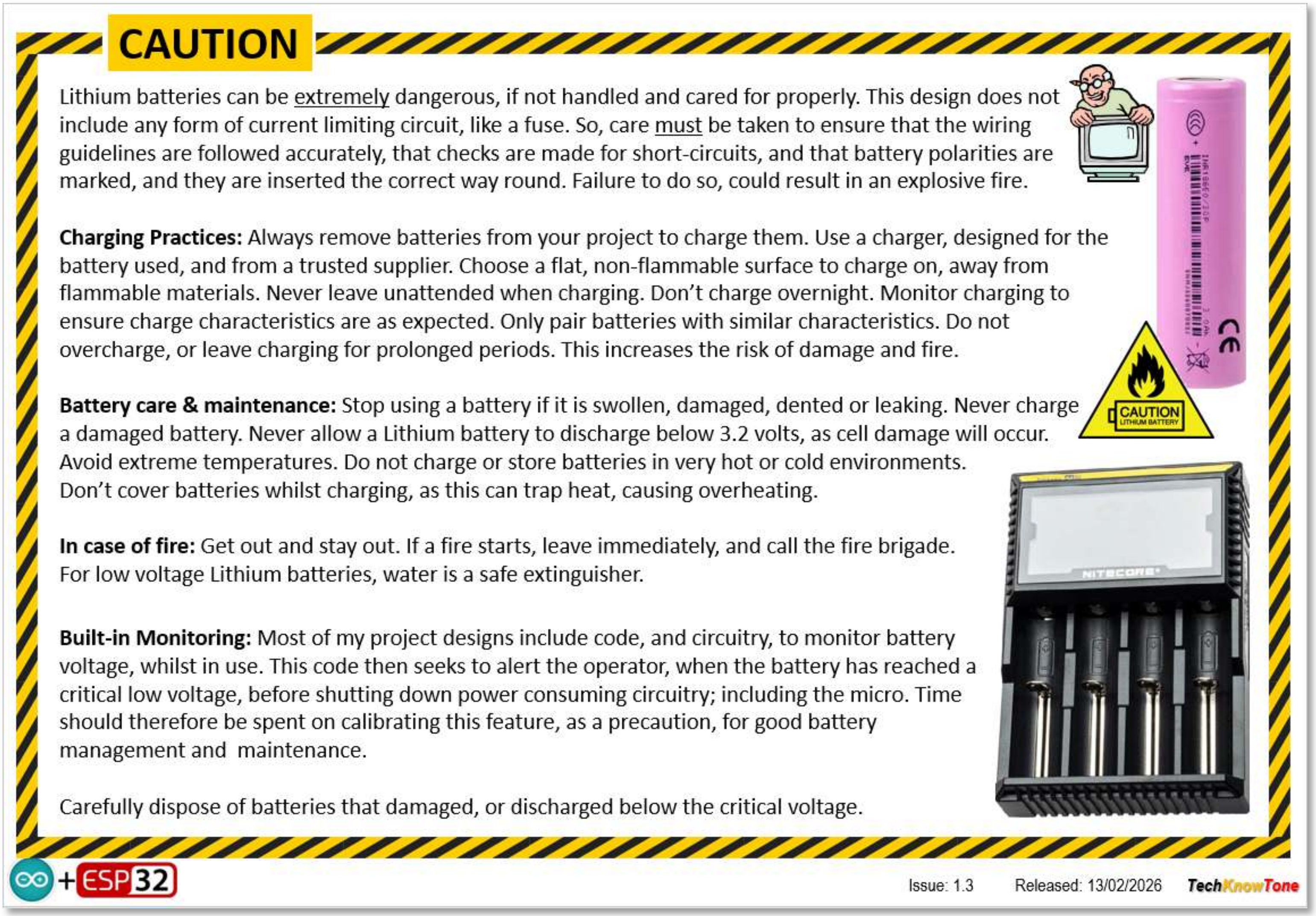
Take Care



It's important to respect the use of this technology.

This design does not include any form of circuit protection, so sloppy wiring, resulting in short circuits, can pose a major hazard.

It does offer good practise, and instructions that should be adhered to. Like the use of a multimeter to check for wiring faults, and the use of a good quality battery charger.



CAUTION

Lithium batteries can be extremely dangerous, if not handled and cared for properly. This design does not include any form of current limiting circuit, like a fuse. So, care must be taken to ensure that the wiring guidelines are followed accurately, that checks are made for short-circuits, and that battery polarities are marked, and they are inserted the correct way round. Failure to do so, could result in an explosive fire.

Charging Practices: Always remove batteries from your project to charge them. Use a charger, designed for the battery used, and from a trusted supplier. Choose a flat, non-flammable surface to charge on, away from flammable materials. Never leave unattended when charging. Don't charge overnight. Monitor charging to ensure charge characteristics are as expected. Only pair batteries with similar characteristics. Do not overcharge, or leave charging for prolonged periods. This increases the risk of damage and fire.

Battery care & maintenance: Stop using a battery if it is swollen, damaged, dented or leaking. Never charge a damaged battery. Never allow a Lithium battery to discharge below 3.2 volts, as cell damage will occur. Avoid extreme temperatures. Do not charge or store batteries in very hot or cold environments. Don't cover batteries whilst charging, as this can trap heat, causing overheating.

In case of fire: Get out and stay out. If a fire starts, leave immediately, and call the fire brigade. For low voltage Lithium batteries, water is a safe extinguisher.

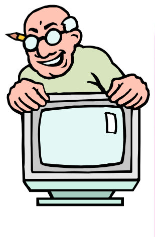
Built-in Monitoring: Most of my project designs include code, and circuitry, to monitor battery voltage, whilst in use. This code then seeks to alert the operator, when the battery has reached a critical low voltage, before shutting down power consuming circuitry; including the micro. Time should therefore be spent on calibrating this feature, as a precaution, for good battery management and maintenance.

Carefully dispose of batteries that damaged, or discharged below the critical voltage.

+ESP32 Issue: 1.3 Released: 13/02/2026 TechKnowTone

CAUTION

Lithium batteries can be extremely dangerous, if not handled and cared for properly. This design does not include any form of current limiting circuit, like a fuse. So, care must be taken to ensure that the wiring guidelines are followed accurately, that checks are made for short-circuits, and that battery polarities are marked, and they are inserted the correct way round. Failure to do so, could result in an explosive fire.



Charging Practices: Always remove batteries from your project to charge them. Use a charger, designed for the battery used, and from a trusted supplier. Choose a flat, non-flammable surface to charge on, away from flammable materials. Never leave unattended when charging. Don't charge overnight. Monitor charging to ensure charge characteristics are as expected. Only pair batteries with similar characteristics. Do not overcharge, or leave charging for prolonged periods. This increases the risk of damage and fire.

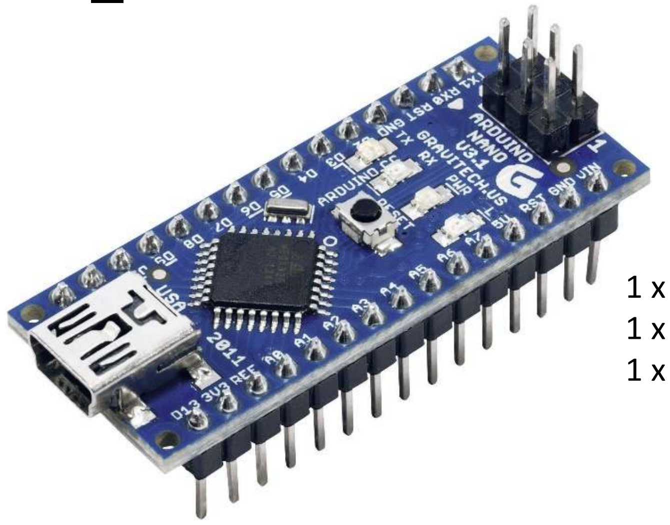
Battery care & maintenance: Stop using a battery if it is swollen, damaged, dented or leaking. Never charge a damaged battery. Never allow a Lithium battery to discharge below 3.2 volts, as cell damage will occur. Avoid extreme temperatures. Do not charge or store batteries in very hot or cold environments. Don't cover batteries whilst charging, as this can trap heat, causing overheating.

In case of fire: Get out and stay out. If a fire starts, leave immediately, and call the fire brigade. For low voltage Lithium batteries, water is a safe extinguisher.

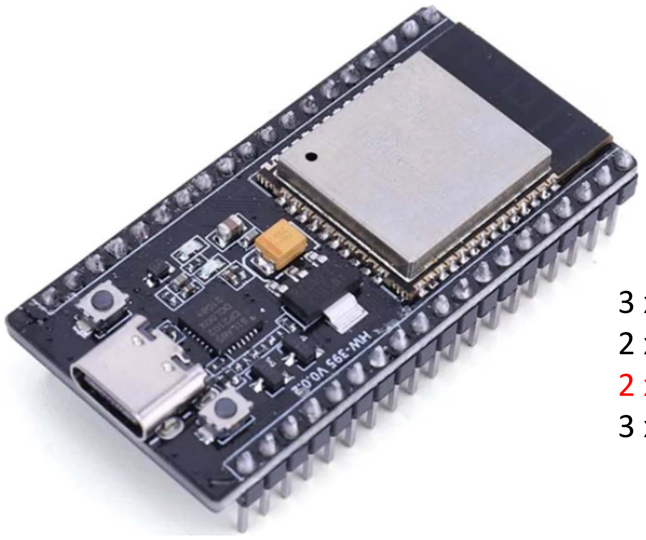
Built-in Monitoring: Most of my project designs include code, and circuitry, to monitor battery voltage, whilst in use. This code then seeks to alert the operator, when the battery has reached a critical low voltage, before shutting down power consuming circuitry; including the micro. Time should therefore be spent on calibrating this feature, as a precaution, for good battery management and maintenance.

Carefully dispose of batteries that damaged, or discharged below the critical voltage.

ARDUINO NANO v ESP32-WROOM-32



1 x UART
1 x I2C
1 x SPI

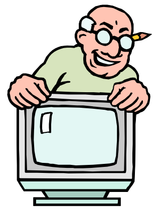


3 x UART
2 x I2C
2 x I2S
3 x SPI

Microcontroller:	ATmega328P (8-bit)
Clock Speed:	16 MHz
Operating Voltage:	5.0 v
VIN:	7 – 12v
Digital I/O:	14 (6 PWM)
Analog Inputs:	8 (10-bit ADC)
Flash Memory:	32 KB (2KB bootloader)
RAM:	2 KB
EEPROM:	1 KB
WiFi:	None

Microcontroller:	Dual core LX6 (32-bit)	**2 x 4 x**
Clock Speed:	240 MHz	**15 x**
Operating Voltage:	2.2 - 3.6 v	
VIN:	5v	
Digital I/O:	22 (16 PWM) + 4 input only	
Analog Inputs:	18 (12-bit, 2 x ADC), 10 touch	**2 x**
Flash Memory:	4 MB (inc. bootloader)	**128 x**
RAM:	520 KB	**260 x**
EEPROM:	512 B	**0.5 x**
WiFi:	802.11 b/g/n + Bluetooth 4.2 BR/EDR/BLE	

Battery Monitoring

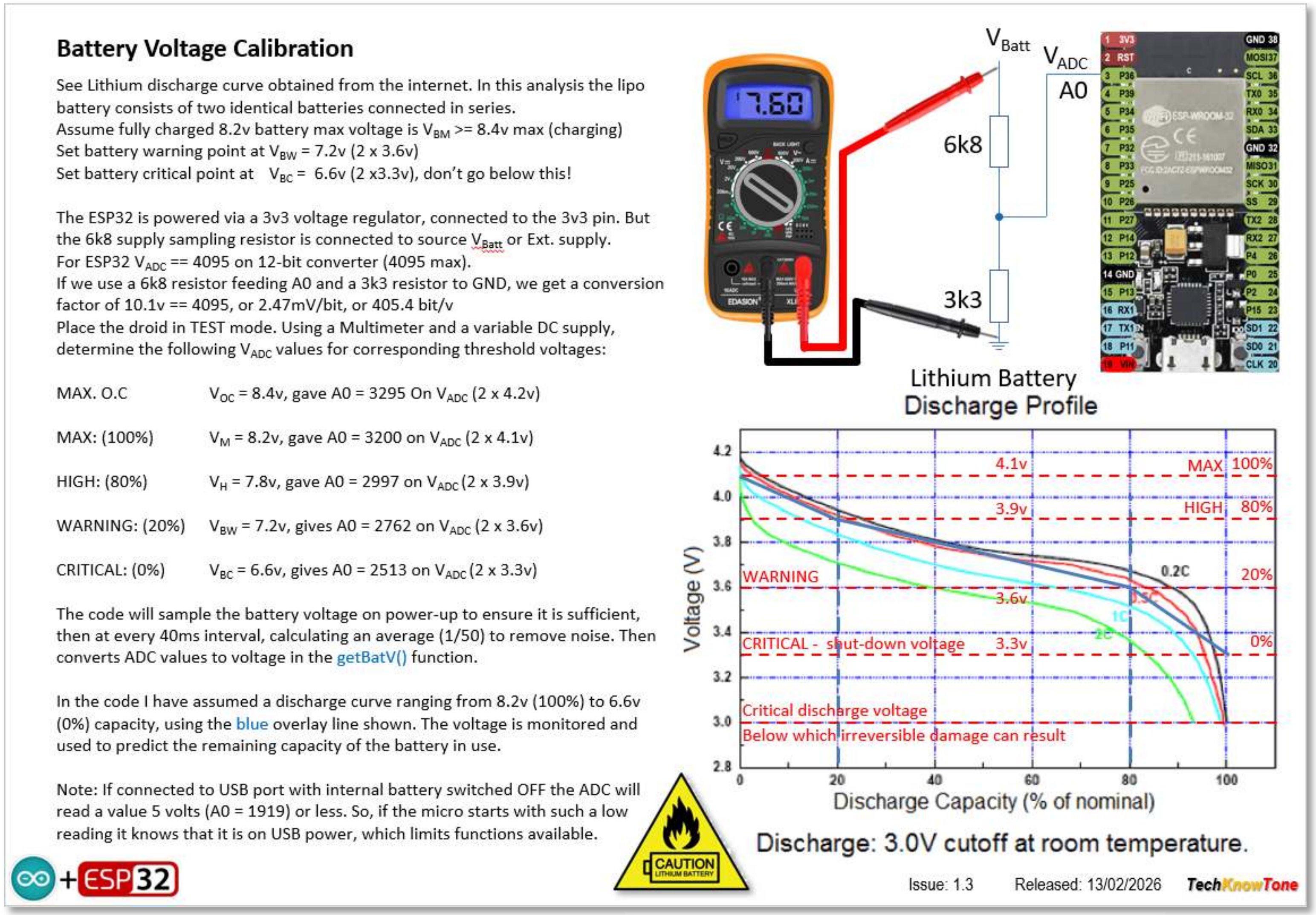


Batteries are based on good chemistry, which is temperature sensitive.

Therefore, good battery monitoring systems measure temperature, voltage and current.

This design takes a more simplistic approach, only relying on voltage measurement, to give an indication of battery health. We calibrate the ESP32’s ADC at key points in a normalised discharge curve.

This gives us a more accurate reading of voltage, and a percentage of capacity based on the discharge curve.



Battery Voltage Calibration

See Lithium discharge curve obtained from the internet. In this analysis the lipo battery consists of two identical batteries connected in series.
Assume fully charged 8.2v battery max voltage is V_{BM} >= 8.4v max (charging)
Set battery warning point at V_{BW} = 7.2v (2 x 3.6v)
Set battery critical point at V_{BC} = 6.6v (2 x3.3v), don’t go below this!

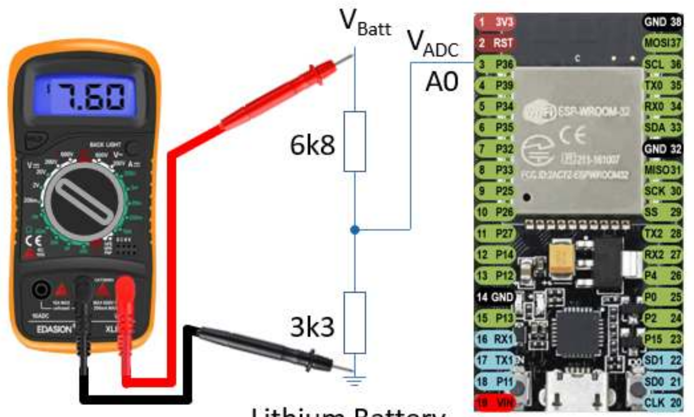
The ESP32 is powered via a 3v3 voltage regulator, connected to the 3v3 pin. But the 6k8 supply sampling resistor is connected to source V_{Batt} or Ext. supply.
For ESP32 V_{ADC} == 4095 on 12-bit converter (4095 max).
If we use a 6k8 resistor feeding A0 and a 3k3 resistor to GND, we get a conversion factor of 10.1v == 4095, or 2.47mV/bit, or 405.4 bit/v
Place the droid in TEST mode. Using a Multimeter and a variable DC supply, determine the following V_{ADC} values for corresponding threshold voltages:

MAX. O.C	V_{OC} = 8.4v, gave A0 = 3295 On V_{ADC} (2 x 4.2v)
MAX: (100%)	V_M = 8.2v, gave A0 = 3200 on V_{ADC} (2 x 4.1v)
HIGH: (80%)	V_H = 7.8v, gave A0 = 2997 on V_{ADC} (2 x 3.9v)
WARNING: (20%)	V_{BW} = 7.2v, gives A0 = 2762 on V_{ADC} (2 x 3.6v)
CRITICAL: (0%)	V_{BC} = 6.6v, gives A0 = 2513 on V_{ADC} (2 x 3.3v)

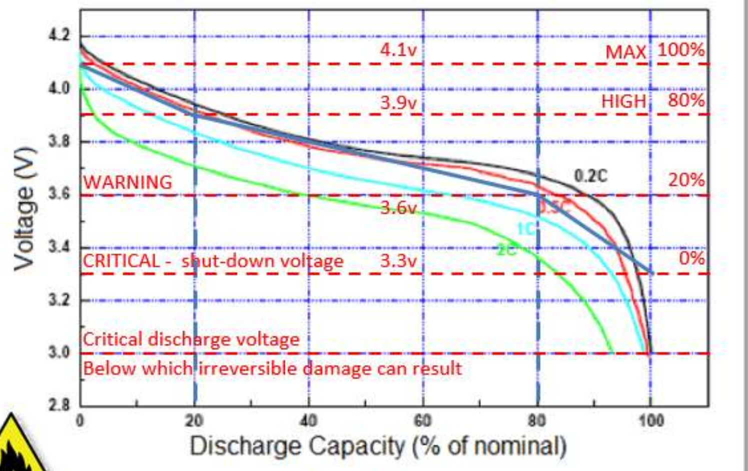
The code will sample the battery voltage on power-up to ensure it is sufficient, then at every 40ms interval, calculating an average (1/50) to remove noise. Then converts ADC values to voltage in the getBatV() function.

In the code I have assumed a discharge curve ranging from 8.2v (100%) to 6.6v (0%) capacity, using the blue overlay line shown. The voltage is monitored and used to predict the remaining capacity of the battery in use.

Note: If connected to USB port with internal battery switched OFF the ADC will read a value 5 volts (A0 = 1919) or less. So, if the micro starts with such a low reading it knows that it is on USB power, which limits functions available.



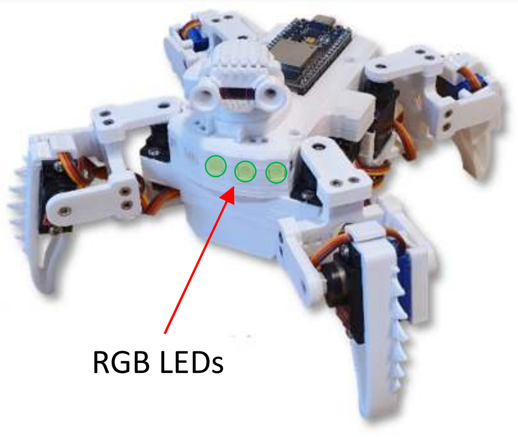
Lithium Battery Discharge Profile



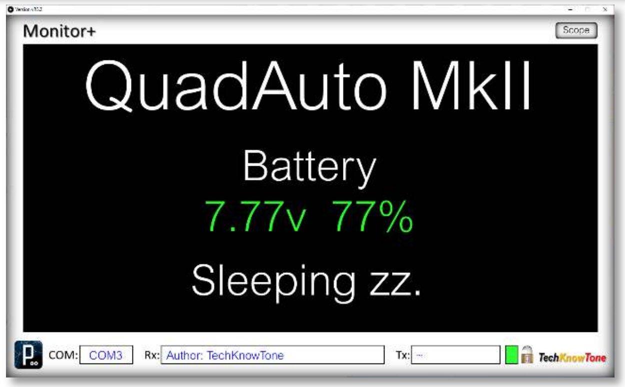
Discharge: 3.0V cutoff at room temperature.

Issue: 1.3 Released: 13/02/2026 TechKnowTone

When powered up the Quadruped indicates battery health, in both its RGB LEDs (colour and number), and through the Monitor+ app, when connected. The Calibration document explains how to measure and enter values in your code.



RGB LEDs



QuadAuto MkII

Battery

7.77v 77%

Sleeping zz.

Battery Voltage Calibration

See Lithium discharge curve obtained from the internet. In this analysis the lipo battery consists of two identical batteries connected in series.
Assume fully charged 8.2v battery max voltage is V_{BM} >= 8.4v max (charging)
Set battery warning point at V_{BW} = 7.2v (2 x 3.6v)
Set battery critical point at V_{BC} = 6.6v (2 x3.3v), don't go below this!

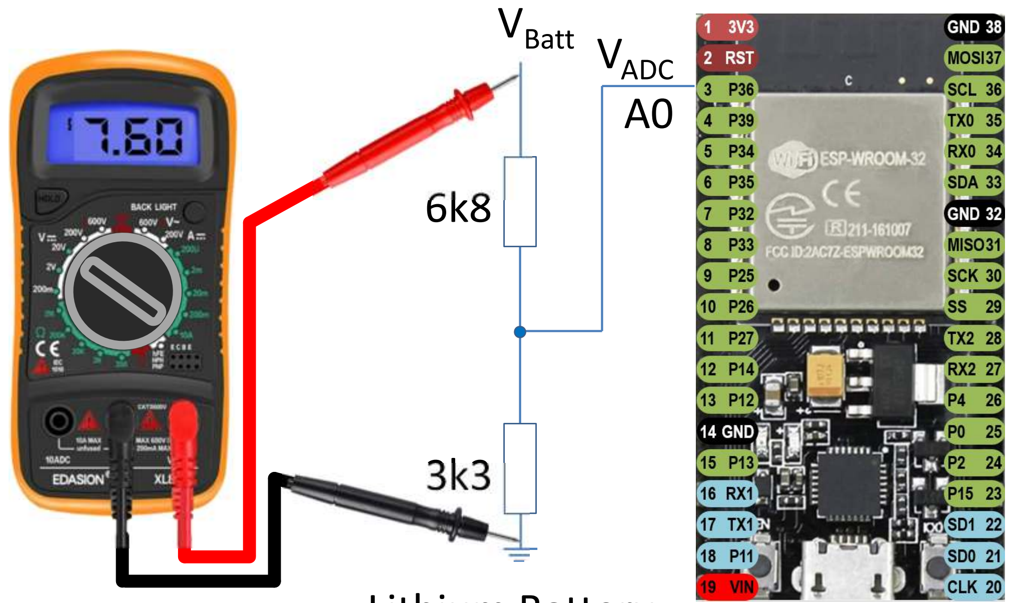
The ESP32 is powered via a 3v3 voltage regulator, connected to the 3v3 pin. But the 6k8 supply sampling resistor is connected to source V_{Batt} or Ext. supply.
For ESP32 V_{ADC} == 4095 on 12-bit converter (4095 max).
If we use a 6k8 resistor feeding A0 and a 3k3 resistor to GND, we get a conversion factor of 10.1v == 4095, or 2.47mV/bit, or 405.4 bit/v
Place the droid in TEST mode. Using a Multimeter and a variable DC supply, determine the following V_{ADC} values for corresponding threshold voltages:

MAX. O.C	V_{OC} = 8.4v, gave A0 = 3295 On V_{ADC} (2 x 4.2v)
MAX: (100%)	V_M = 8.2v, gave A0 = 3200 on V_{ADC} (2 x 4.1v)
HIGH: (80%)	V_H = 7.8v, gave A0 = 2997 on V_{ADC} (2 x 3.9v)
WARNING: (20%)	V_{BW} = 7.2v, gives A0 = 2762 on V_{ADC} (2 x 3.6v)
CRITICAL: (0%)	V_{BC} = 6.6v, gives A0 = 2513 on V_{ADC} (2 x 3.3v)

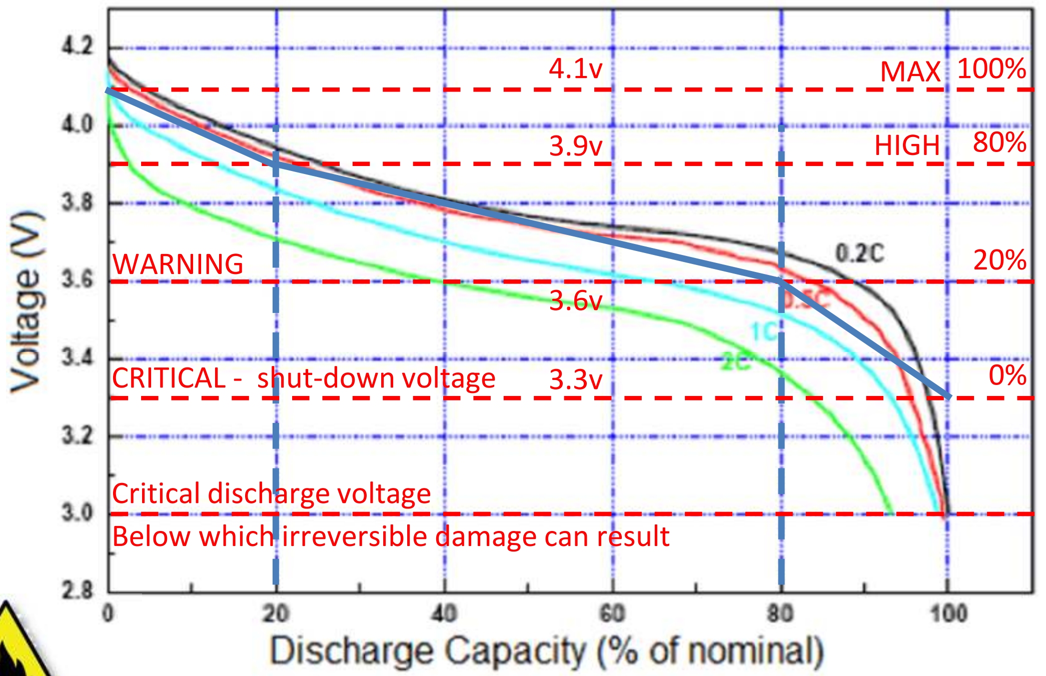
The code will sample the battery voltage on power-up to ensure it is sufficient, then at every 40ms interval, calculating an average (1/50) to remove noise. Then converts ADC values to voltage in the getBatV() function.

In the code I have assumed a discharge curve ranging from 8.2v (100%) to 6.6v (0%) capacity, using the blue overlay line shown. The voltage is monitored and used to predict the remaining capacity of the battery in use.

Note: If connected to USB port with internal battery switched OFF the ADC will read a value 5 volts (A0 = 1919) or less. So, if the micro starts with such a low reading it knows that it is on USB power, which limits functions available.

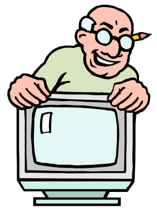


Lithium Battery
Discharge Profile



Discharge: 3.0V cutoff at room temperature.

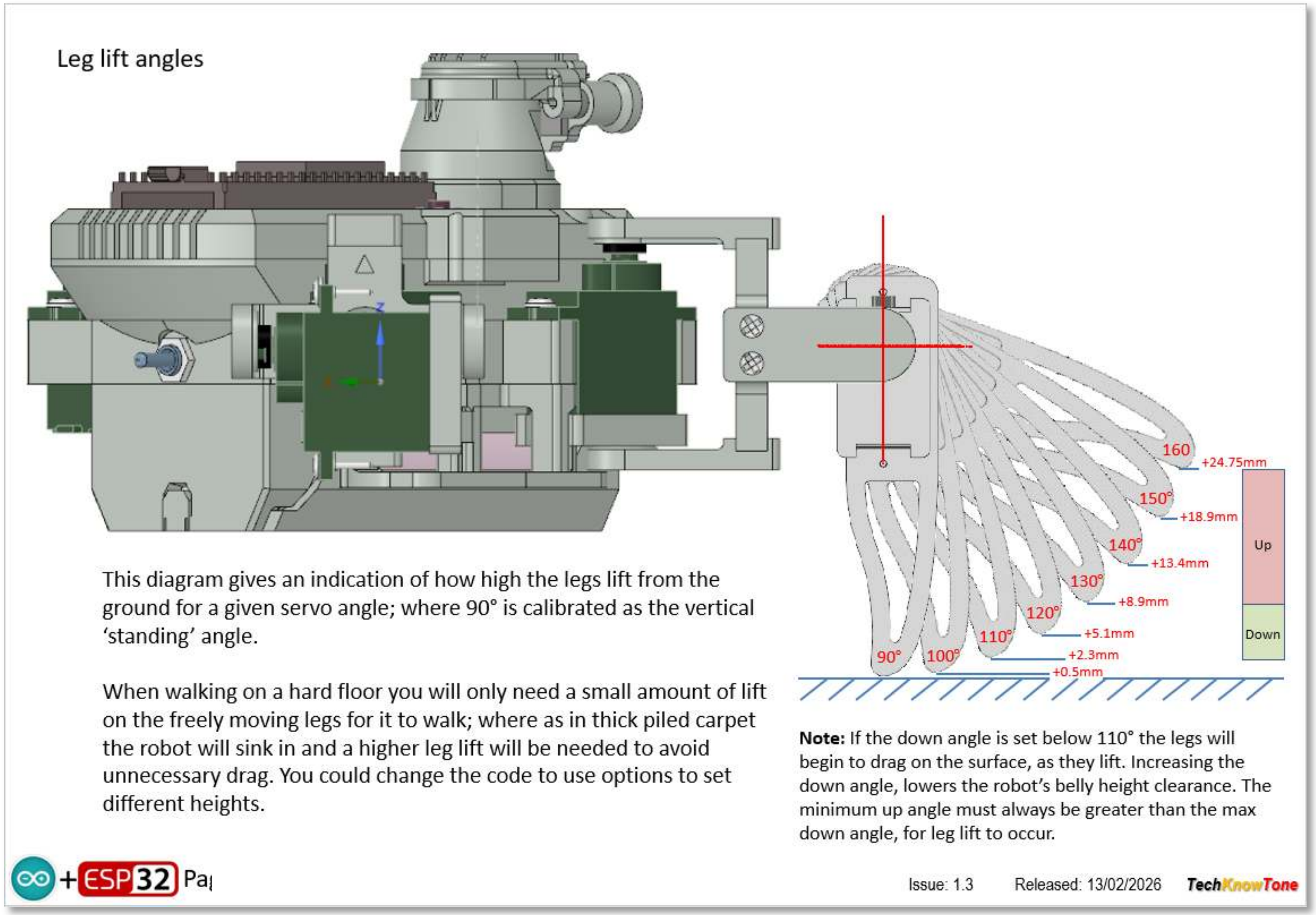
Leg Lift Angles



As the Quadruped only uses two servo motors in each leg, these effectively act as a shoulder joint. A leg can be rotated around the body, and swung out. But it can not be lifted, like a normal leg.

In order to walk, the robot needs to lower its body height, by swinging the legs out slightly. Then it can effectively raise them, by swinging them upwards.

On a flat surface this works fine, and there is no need to raise them too heigh when walking. But on a carpet, the legs need to be raised more. Code options, depending on the Wii controller used, allow you to raise the legs when needed, at low speed.



Leg lift angles

This diagram gives an indication of how high the legs lift from the ground for a given servo angle; where 90° is calibrated as the vertical 'standing' angle.

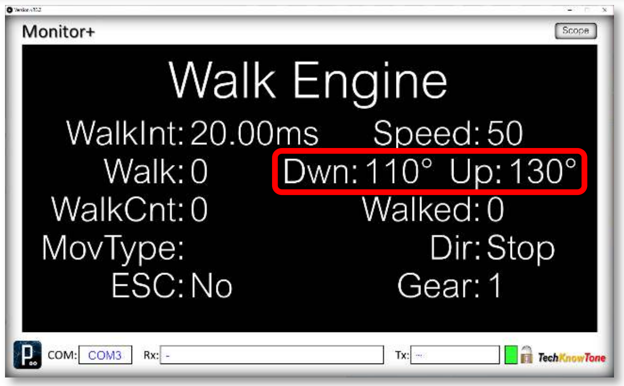
When walking on a hard floor you will only need a small amount of lift on the freely moving legs for it to walk; where as in thick piled carpet the robot will sink in and a higher leg lift will be needed to avoid unnecessary drag. You could change the code to use options to set different heights.

Note: If the down angle is set below 110° the legs will begin to drag on the surface, as they lift. Increasing the down angle, lowers the robot's belly height clearance. The minimum up angle must always be greater than the max down angle, for leg lift to occur.

+ESP32 Pa

Issue: 1.3 Released: 13/02/2026 TechKnowTone

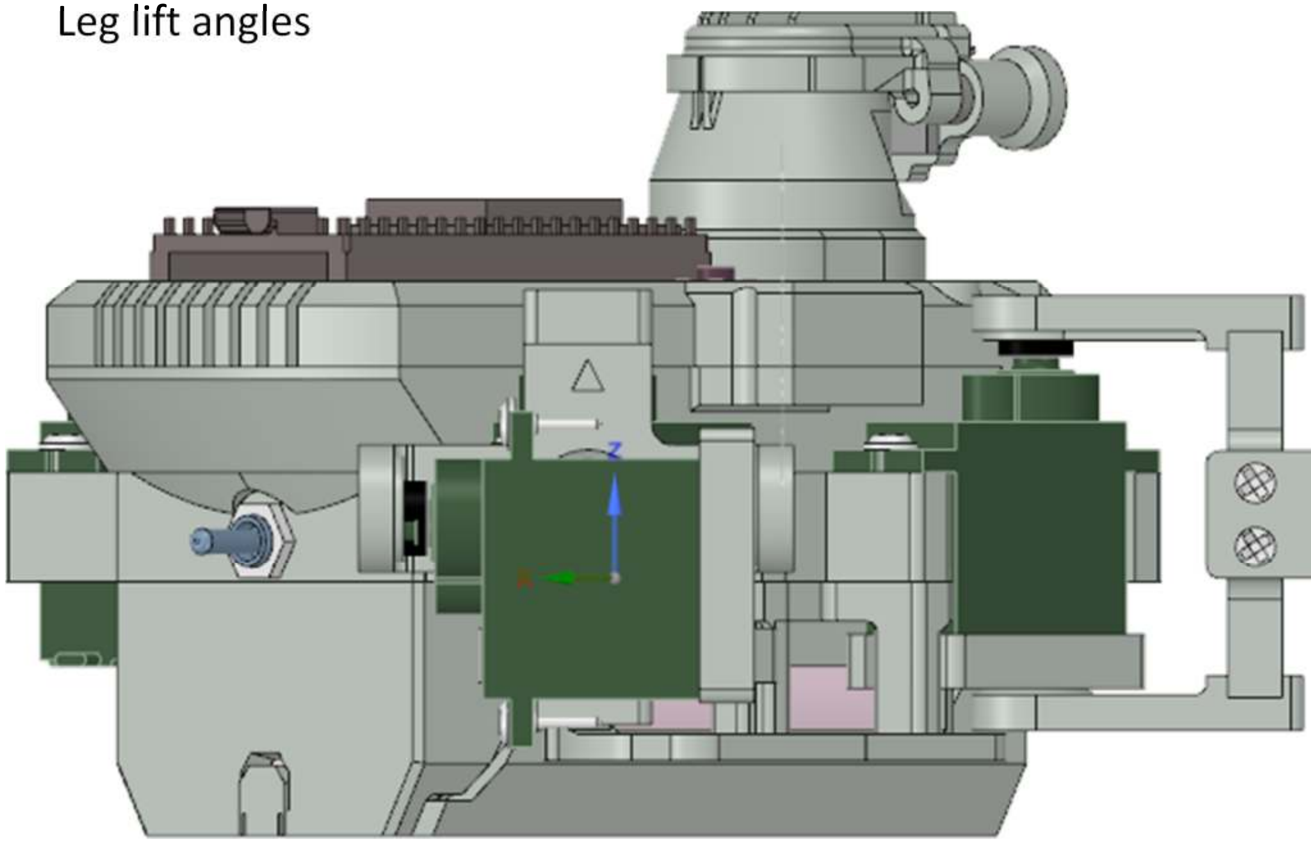
When using the Monitor+ app you can view the angles being used for leg down and leg up. For flat surfaces these are 110° and 130° respectively.

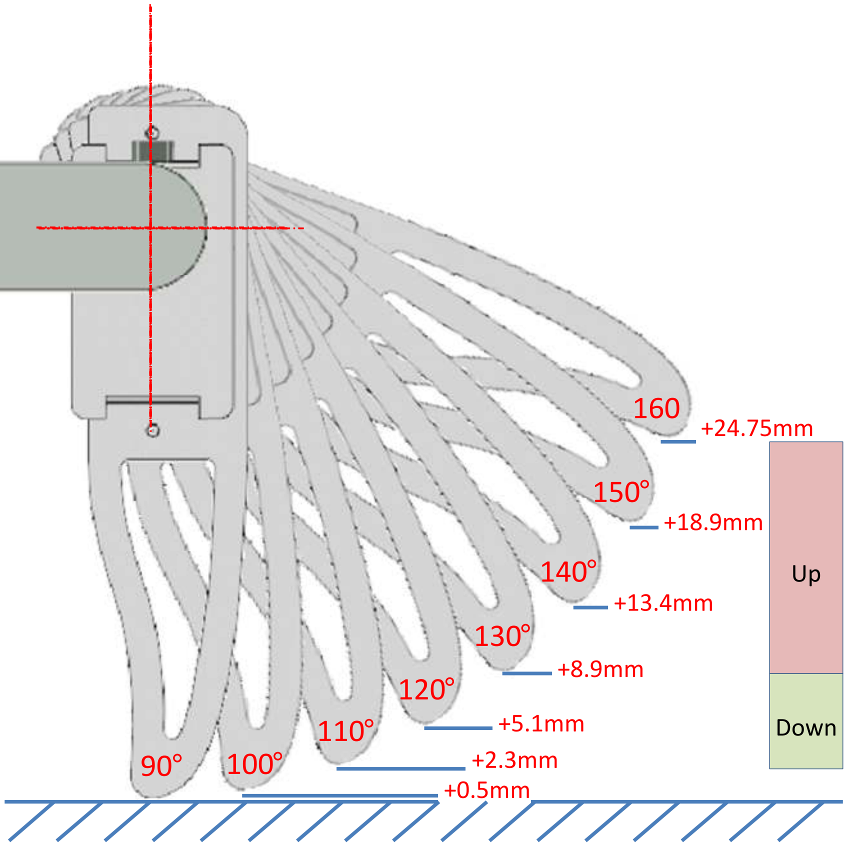


```
Walk Engine
WalkInt: 20.00ms      Speed: 50
   Walk: 0      Dwn: 110°  Up: 130°
WalkCnt: 0           Walked: 0
MovType:                Dir: Stop
    ESC: No            Gear: 1
```

COM: COM3 Rx: - Tx: -

Leg lift angles





This diagram gives an indication of how high the legs lift from the ground for a given servo angle; where 90° is calibrated as the vertical 'standing' angle.

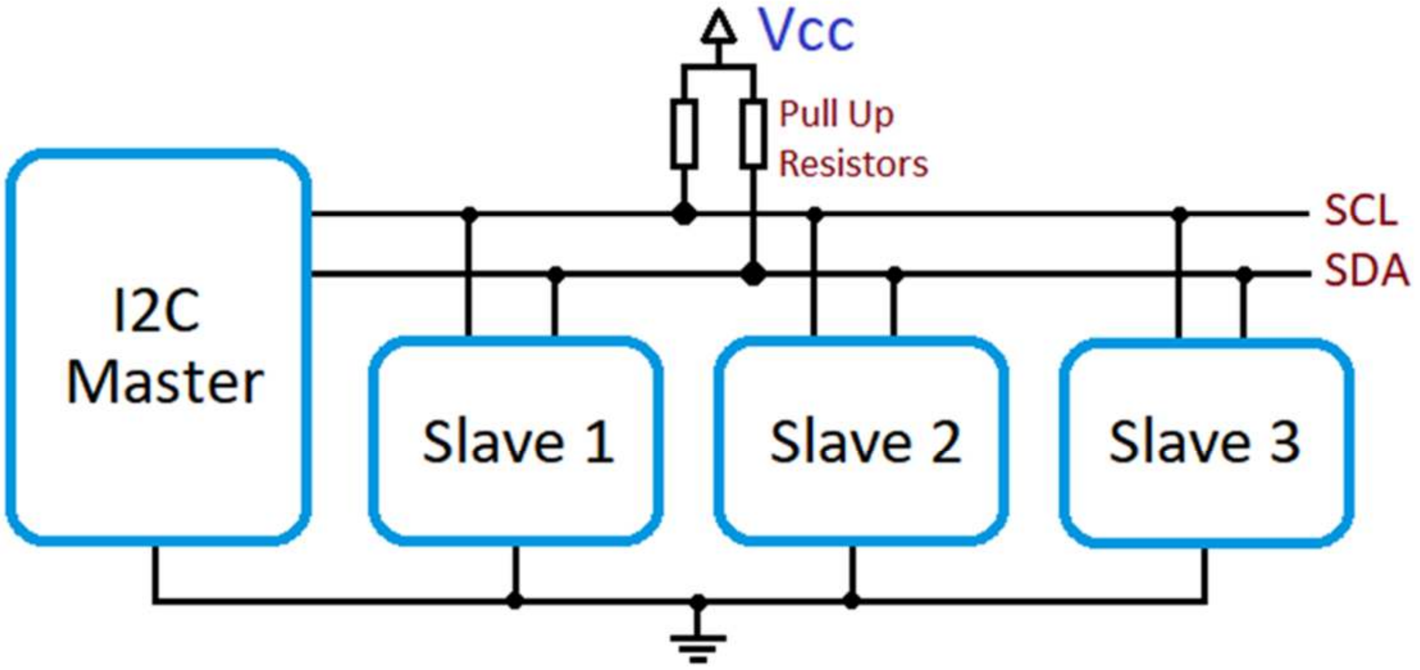
When walking on a hard floor you will only need a small amount of lift on the freely moving legs for it to walk; where as in thick piled carpet the robot will sink in and a higher leg lift will be needed to avoid unnecessary drag. You could change the code to use options to set different heights.

Note: If the down angle is set below 110° the legs will begin to drag on the surface, as they lift. Increasing the down angle, lowers the robot's belly height clearance. The minimum up angle must always be greater than the max down angle, for leg lift to occur.

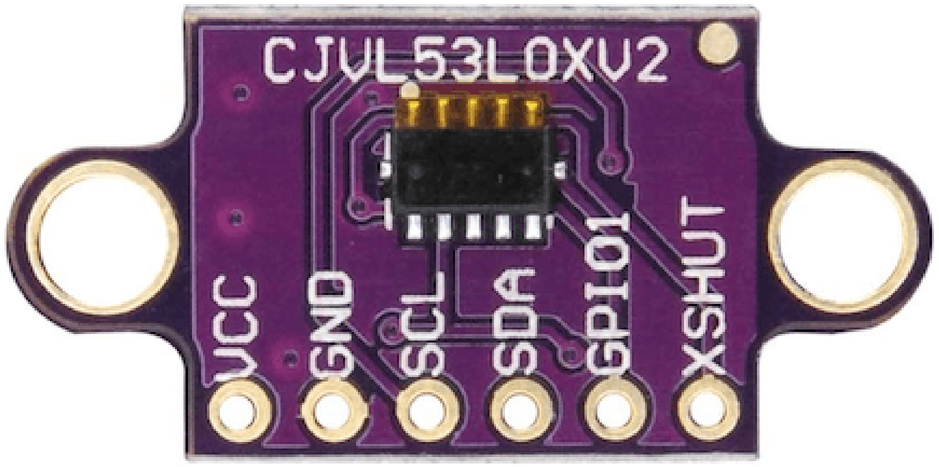
Walking Basics

WS2812B RGB LEDs

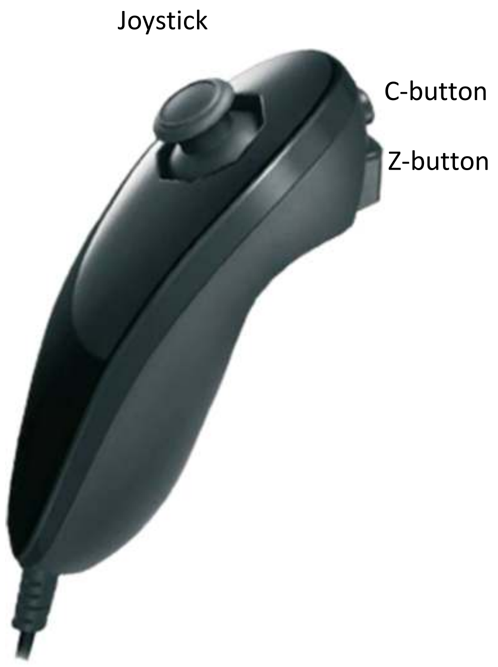
I2C Data Bus


Vcc
Pull Up
Resistors
SCL
SDA
I2C
Master
Slave 1
Slave 2
Slave 3

VL53L0X Laser Range Finder

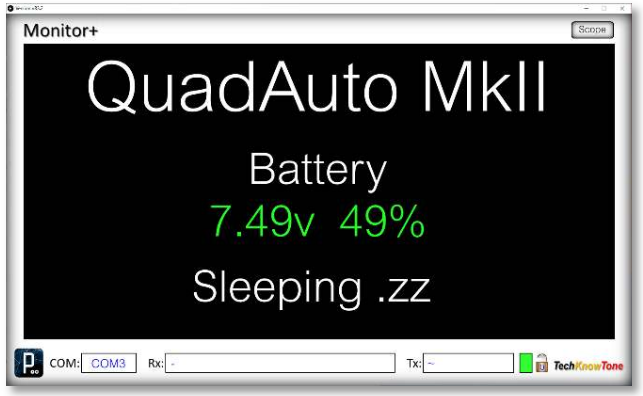


Wii Controllers – I2C



ESPNOW 2.4 GHz WiFi

Monitor+ App

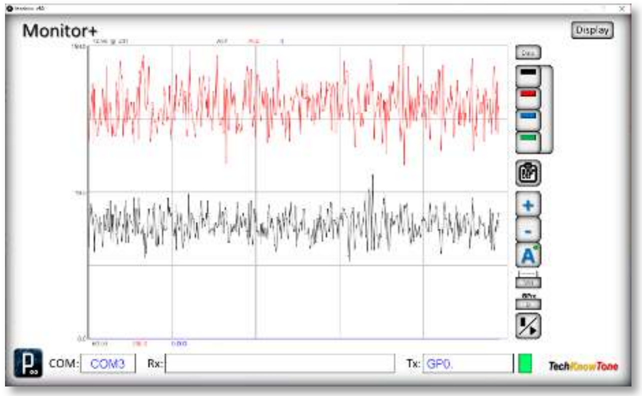


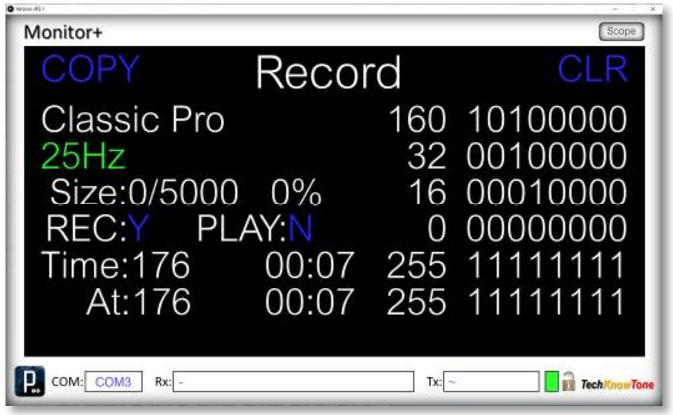
QuadAuto MkII

Battery

7.49v 49%

Sleeping .zz





```
COPY          Record           CLR
Classic Pro              160  10100000
25Hz                      32  00100000
 Size:0/5000   0%         16  00010000
 REC:Y    PLAY:N           0  00000000
Time:176       00:07     255  11111111
   At:176      00:07     255  11111111
```