

Self Balancing Car with Wemos D1 R32

Here are my notes on this project. We are using the [Keystudio KS0193](#) two wheel self-balancing car kit to start with. Reference it for documentation on the car. These notes concern the modifications I did to it. More documentation is available on [fweb](#).

The Arduino Uno car uses is not good enough for the Control class to use unless really in a pinch. It isn't fast enough to debug the Kalman filter program and needs more memory. The Wemos D1 R32 uses the same form factor as the Uno and works with some of the code. The [Keystudio KS0377](#) shield was retained. The following changes are needed:

- The encoders need to be quadrature. There are quadrature encoders on the hardware but not in the Keystudio software. We added that using the [ESP32Encoder library](#). The encoder only demo program is called Encoder.ino,
- The ATMEGA328 PWM code had to be modified to use the ESP32. We chose to use the easier LEDC PWM hardware instead of the fancy PWM code that has punch through (short from V_{DD} to GND) because the TB6612FNG has that feature in the hardware and it is used in the KS0377 arduino shield used to drive the motors etc. The schematic shows what pins to connect, and the D1 R32 uses 'IO25' to denote 'IO25' which I think could be called 25 as well. The pins had to be listed in the call to the encoder library. See the code and the comparison pinout for the original Uno and the D1 R32 on the next page.
- The motor drive had to be connected in code. The simple test code for the motors only is called WemosD1KeyCarMotorDrive. Then this was integrated with the encoder code above to create WemosD1KeyCarMotorDriveEncoder which shows the speed and position of the wheels (in some as yet unknown units) and you can plot it using the Arduino plot window, and see the data in the Serial Monitor.
- The MPU-6050 uses I2C, and is wired to IO36 and IO39 on the D1 R32, but [those pins are for input only, so cannot be remapped to be used for the I2C lines](#). Remapping is discussed [here](#). Therefore, jumpers must extend between IO39 pin to SCL and IO36 and SDA respectively. On the Uno these were the same were wired together to the same pins. There are holes so these could be soldered together on the shield if desired. The library used by Keystudio and retained is the [MPU6\[50 library](#). The software to test this is the MPU_6050_ESP32_Test. The program that integrates the MPU-6050 with the rest of the things is called WemosD1KeyCarMotorEncodersAccel.
- You can see the angle and Kalman calculated angle converge as well as the gyro_x and the Kalman filtered gyro_x using the Kalman_Test program and the serial plotter.

This may be enough software to give students to let them make software to control the car.

Pinout:

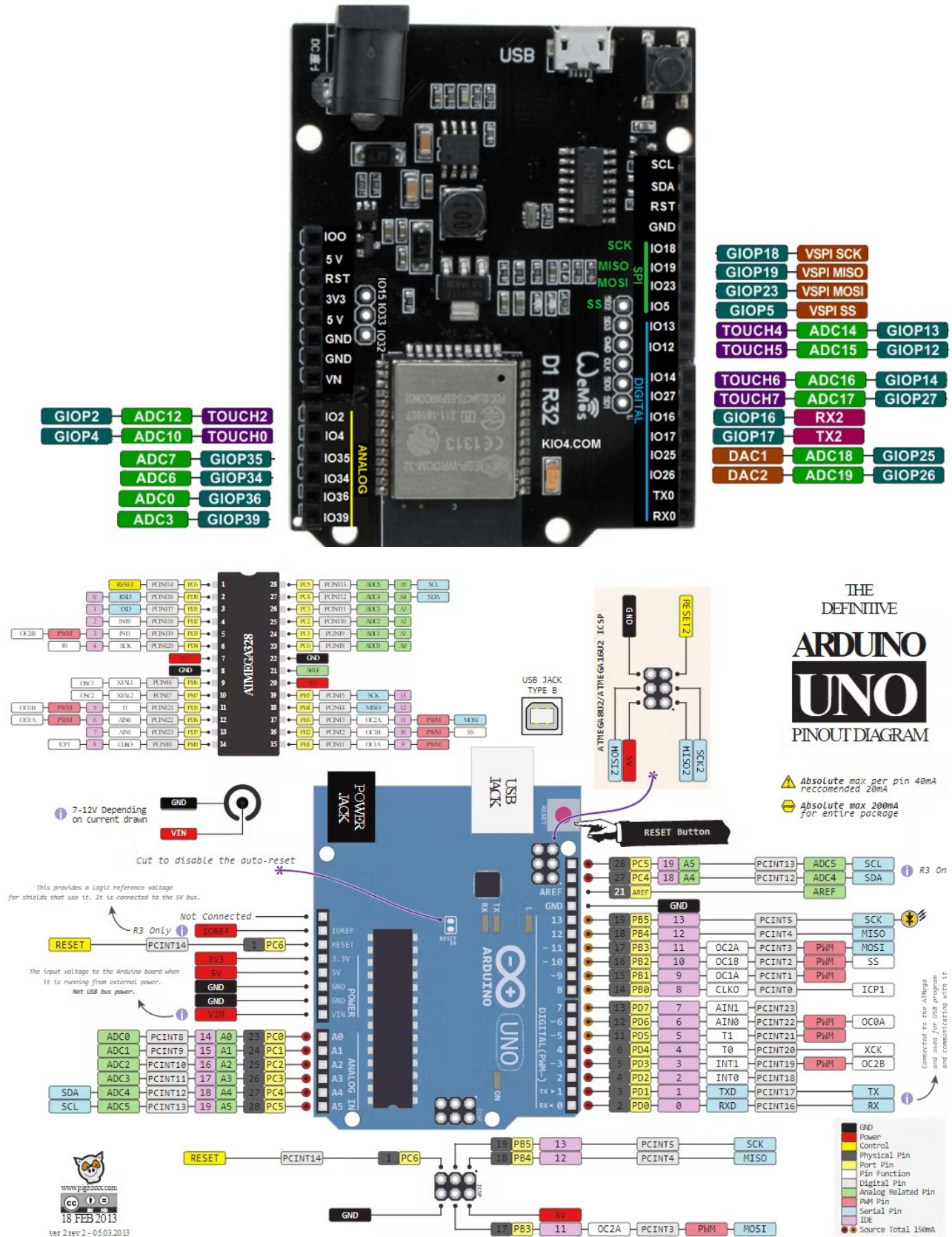


Figure 1: Arduino Uno and Wemos D1 R32 Pin-Outs for Comparison

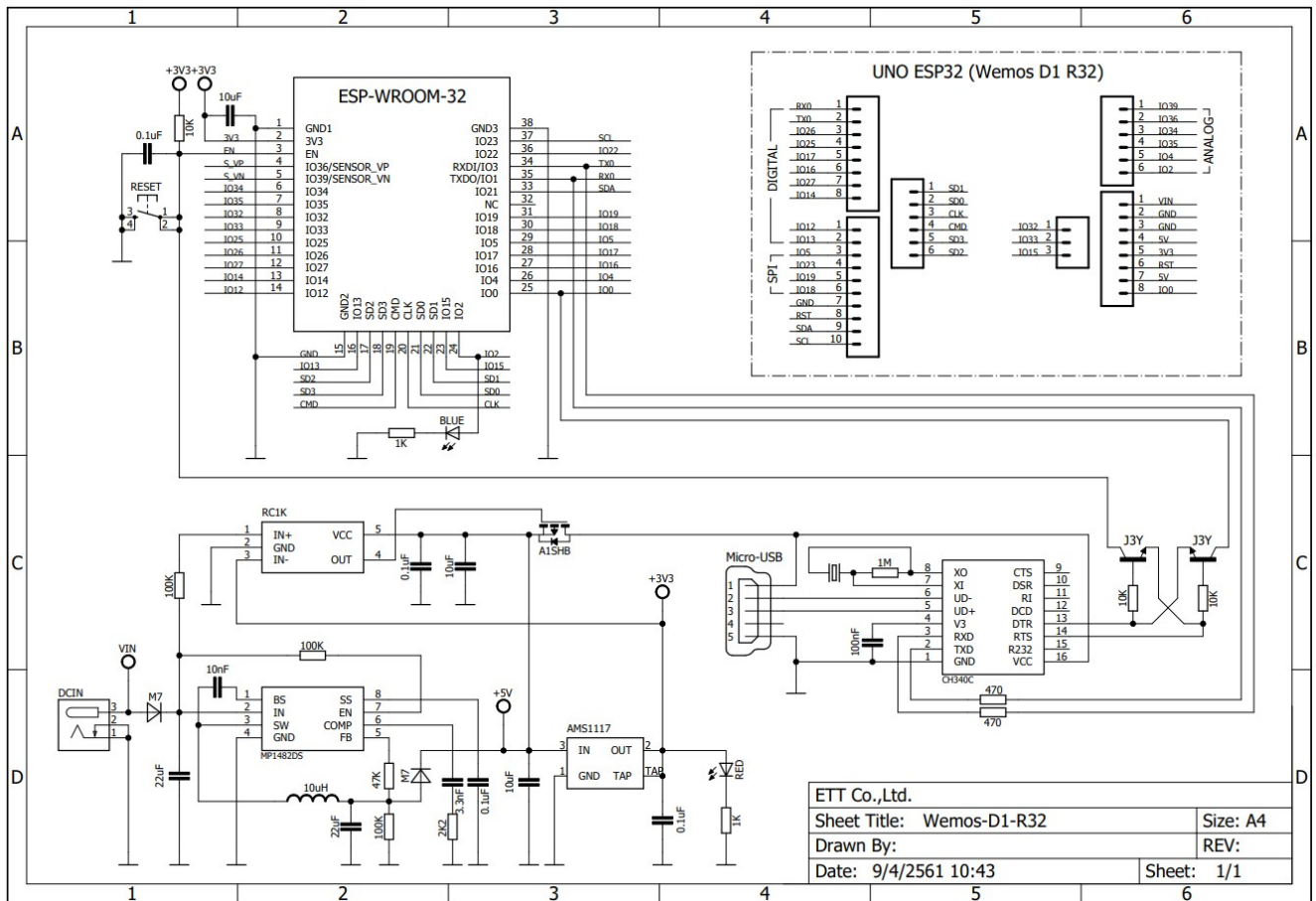
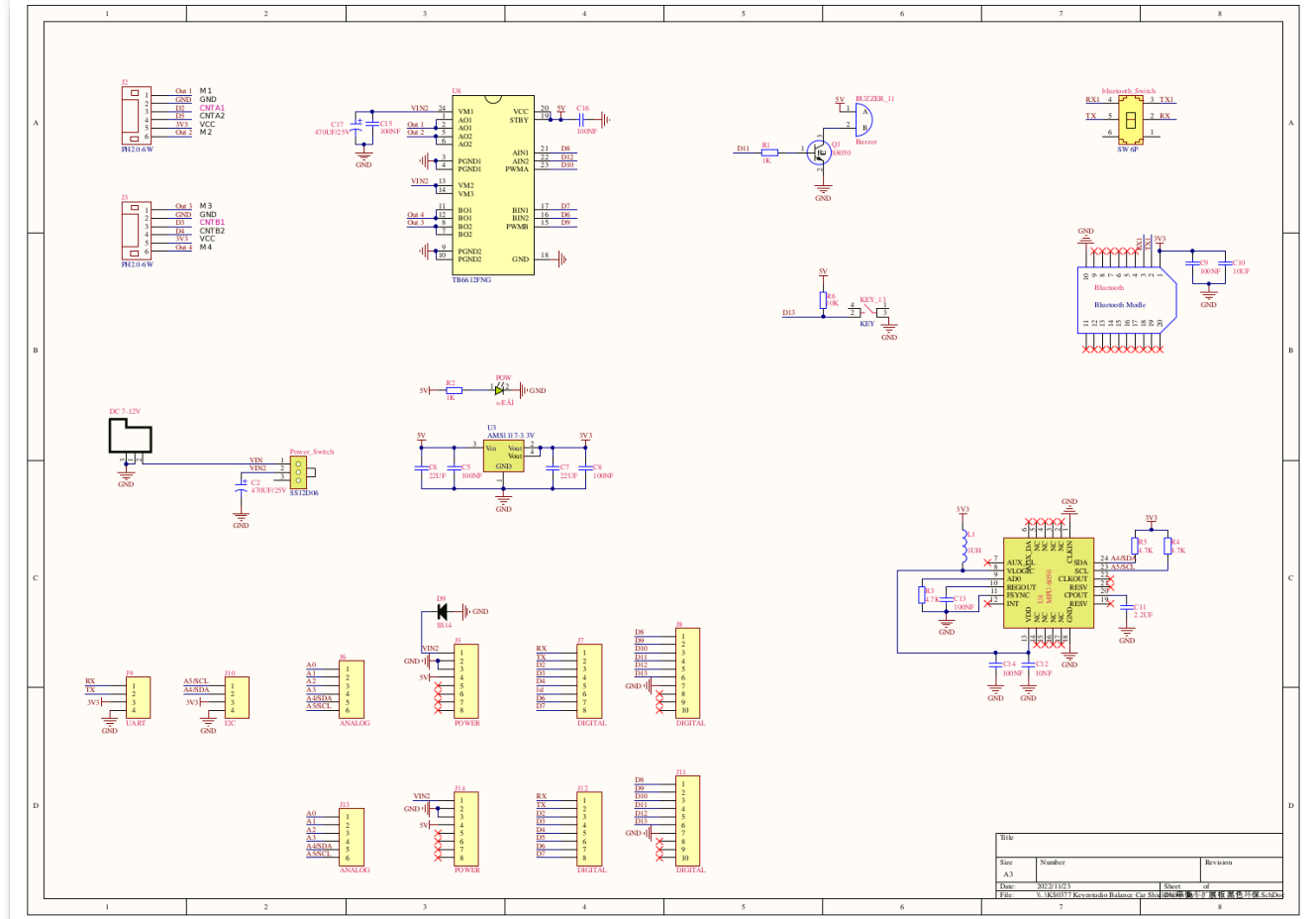


Figure 2: KS0377 Shield and Wemos D1 R32 Schematic Diagram

