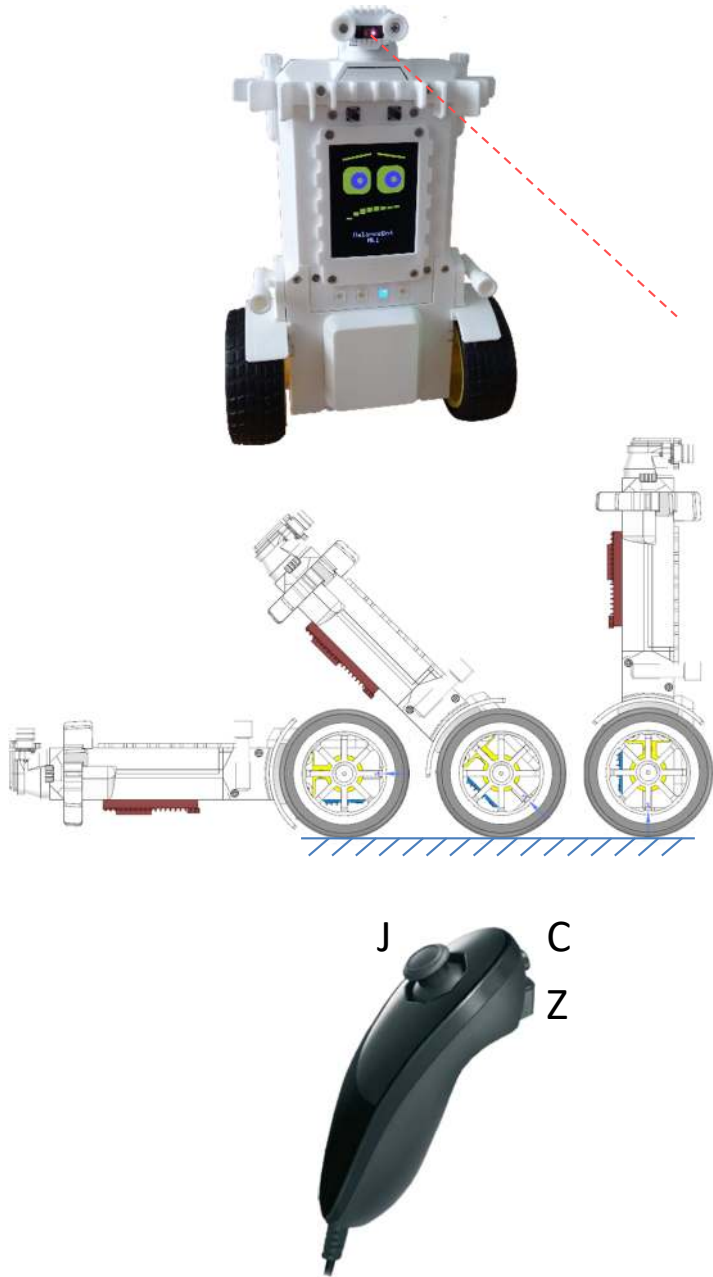


BalanceBot Mk1– Demo Functions



Tech:

- ESP32 microcontroller, 240 MHz, 32-bit, 2-core
- Wii controller + WEMOS Wi-Fi link remote control
- 2.4 GHz Wi-Fi link range ~100 m (ESP-NOW)
- 2 x 6v DC motors + 2 x slot encoders
- MPU6050 3-axis gyros and accelerometers (250Hz)
- VLX53L1X laser range finder in pan head
- 2.4" colour TFT display
- 4 x WS2812B RGB LED lights
- 7.5v 3000mAh battery
- 3-D printed construction
- A custom mounting plate aids internal wiring

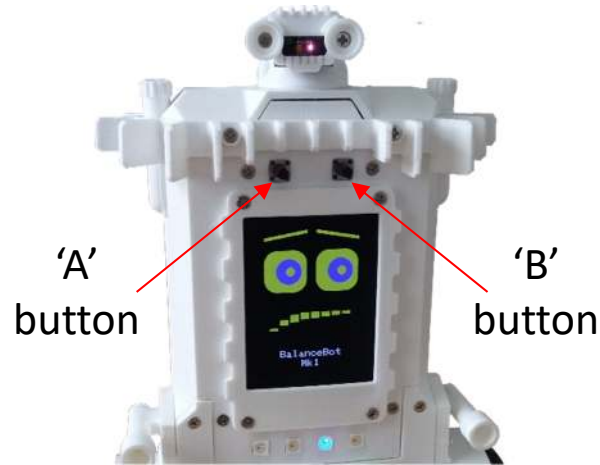
Features:

- Gyro calibration on reset (or on demand)
- Safe start, angle displayed and topple stop
- Animated face when going live and active
- Sleeps when unattended, wakes when moved
- Controlled from Wii controller over Wi-Fi, using ESP-NOW
- Robot data returned over coms link
- Nunchuk C/Z buttons selects speed settings (5 speeds)
- Tilt-back head for battery access & charging

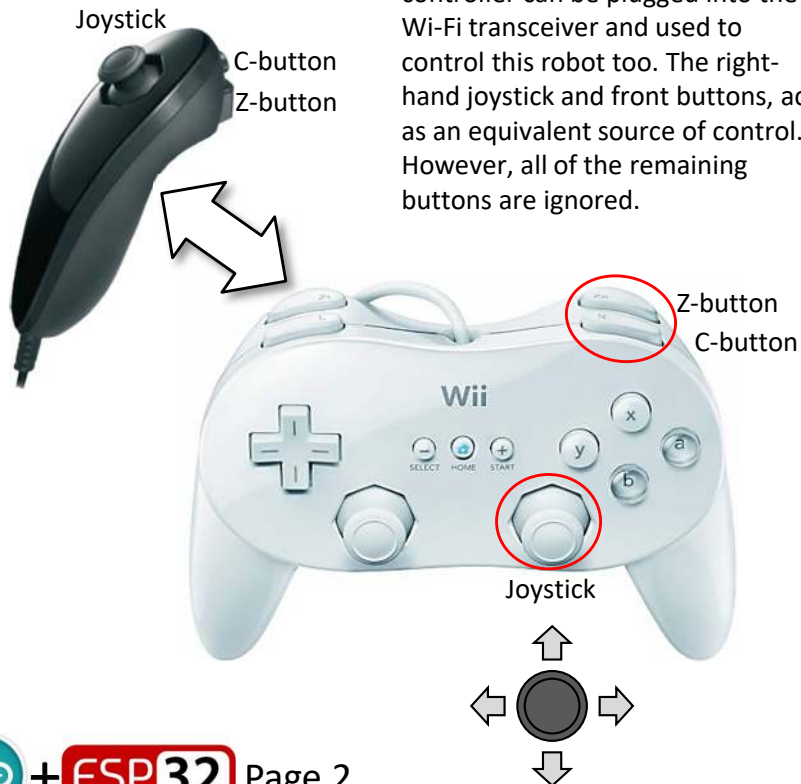
Enhancements (TBD):

- Responds to hand gestures
- Autonomous behaviour

BalanceBot Mk1– Button Functions



Note that a Wii Classic Pro controller can be plugged into the Wi-Fi transceiver and used to control this robot too. The right-hand joystick and front buttons, act as an equivalent source of control. However, all of the remaining buttons are ignored.



Operation:

- Place the robot on its stand, and then switch on.
- Allow it a few seconds to calibrate the MPU6050 gyro offsets.
- The LEDs will flash random colours during this process.
- If left in this position, it should shortly soon go to sleep.
- When sleeping, if moved, it will wake up.
- If LEDs are green, press a button to connect to WiFi. If successful, then blue.
- Lie it on its back, to initiate self-balancing sequence.
- After a few seconds the red LED will begin to flash, 'READY'.
- As you raise the robots head towards the vertical, the red LED will flash more rapidly and the facial expressions will exhibit alarm.
- The angle of the robot is also displayed beneath the animated face.
- Once you reach the vertical position, 0° the motors will engage.
- Whilst balancing the contents of the lower display is as set via button switches during the 'READY' state, or via a Windows Monitor+ app.
- Tilting or toppling the robot will remove power from the drive motors.

Buttons:

- A long press on 'B' will initiate self-reset and calibration mode.
- When on it's back, in the 'READY' state:
 - button 'A' selects a range of parameters to be displayed in the lower screen, like PID variables and slot counter values.
 - Button 'B' selects different modes of operation. In this release only the 'BALANCE' mode is fully implemented.

Nunchuk:

- When balancing joystick 'J' controls turning and forward motion.
- Button 'C' increases the drive and steering speeds (1 – 5).
- Button 'Z' decreases the drive and steering speeds (5 – 1).

BalanceBot MK1 – Code Info.

Wireless Controller:

- WEMOS ESP8266 4MB, 80 MHz, 32-bit, 1-core
- Libraries: Wire.h, SoftwareSerial.h
- Wire.h needed for I2C bus, Nunchuk controller
- Reads for Wii controller demands.
- LED flash warning if not present.
- Main loop runs every 40ms (25 Hz. hardware timer)
 - Reads Nunchuk registers (six 8-bit registers)
 - Sets corresponding bits in a data byte
 - In critter mode create data frame
 - Send data to Wi-Fi transceiver
 - Report data if enabled or transfer Rx data
 - Check battery voltage
 - Flash LED
 - once every second - I'm ON
 - Double flash if in Critter mode
 - Flash when sending data

BalanceBot:

- ESP32 4MB, 240 MHz, 32-bit, 2-core
- Libraries: Adafruit_GFX.h, Adafruit_ILI9341.h, Wire.h, ESP32Servo.h, FastLED.h, SparkFun_VL53L1X.h
- Wire.h needed for I2C bus, 3-axis accelerometer/gyro
- Take 1000 gyro readings to allow offsets to settle.
- Main loop runs every 4ms (250 Hz. hardware timer)
 - Checks for SAFE condition first, then indicates READY
 - In READY state flash LEDs to indicate angle
 - Once vertical enter BALANCE, safe mode 4.
 - Read gyro & accelerometer at 250 Hz.
 - Check for falling over – go SAFE, motors off.
 - Receives ESP-NOW 2.4GHz wireless data and respond
 - Uses joystick X-data for steering
 - Uses joystick Y-data for forward/backward drive
 - Z-gyro controls rotation.
 - Perform PID calculation at 250 Hz.
 - Reads buttons for speed change (125 Hz).
 - Check and display battery voltage value.
 - Performs LED task(s), depending on mode.
- Proportional PWM drive to DC motors.