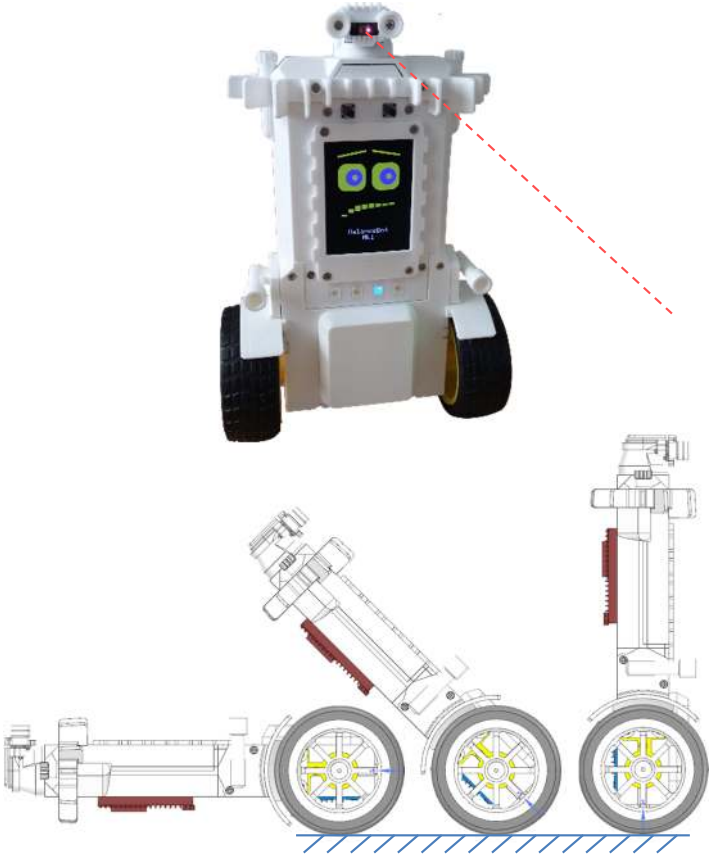
BalanceBot Mk1– Demo Functions



Tech:

- ESP32 microcontroller, 80 MHz, 32-bit, 2-core
- Nunchuk + 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
- 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

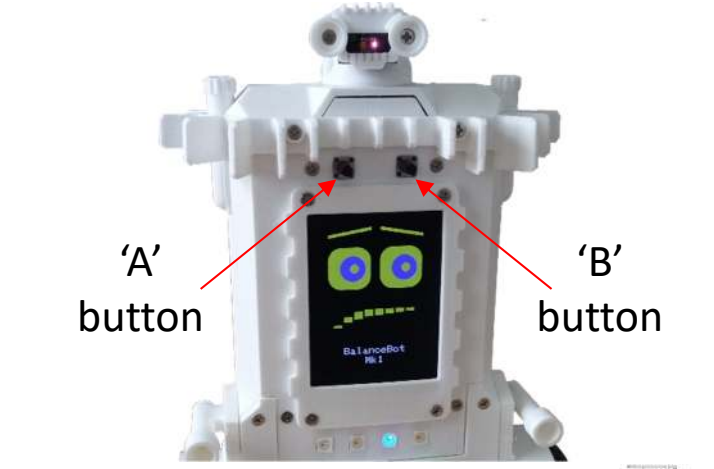
Features:

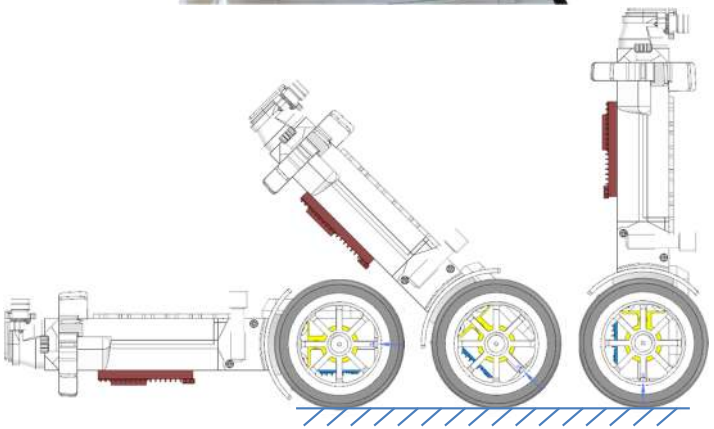
- Gyro calibration on reset (or demand)
- Safe start, angle displayed and topple stop
- Animated face when going live and active
- Sleeps when unattended, wakes when moved
- Controlled from Nunchuk over Wi-Fi
- Robot data returned over coms link
- Nunchuk C/Z buttons selects speed settings
- Tilt-back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

Enhancements (TBD):

- Responds to hand gestures
- Autonomous behaviour

BalanceBot Mk1– Demo Functions





Operation:

- Place the robot on its stand and then switch on.
- Allow is a few seconds to calibrate the MPU6050 gyros.
- The LEDs will flash random colours during this process.
- If left in this position, it should soon go to sleep.
- When sleeping, if moved, it will wake up.

- Lie it on its back, to initiate self-balancing.
- 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 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.
- 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.
- Button ‘Z’ decreases the drive and steering speeds.

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
- Checks for Nunchuk responses
- LED flash warning if not present
- Checks for user mode – BalanceBot v Critter RC
- Main loop runs every 40ms (25 Hz. hardware timer)
    - Reads Nunchuk registers (5)
    - 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, 80 MHz, 32-bit, 2-core
- Libraries: Wire.h
- Wire.h needed for I2C bus, 3-axis accelerometer/gyro
- Take 500 gyro readings to allow it 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 mode
    - Read gyro & accelerometer
    - Check for falling over – go SAFE
    - Receive wireless data and respond
        - Use joystick X-data for steering
        - Use joystick Y-data for forward/backward
        - Z-gyro controls rotation
        - Perform PID calculation
        - Check buttons for speed change
    - Check battery voltage is 'good'
    - Performs LED task(s)
- Hardware timer interrupt clocks stepper motors