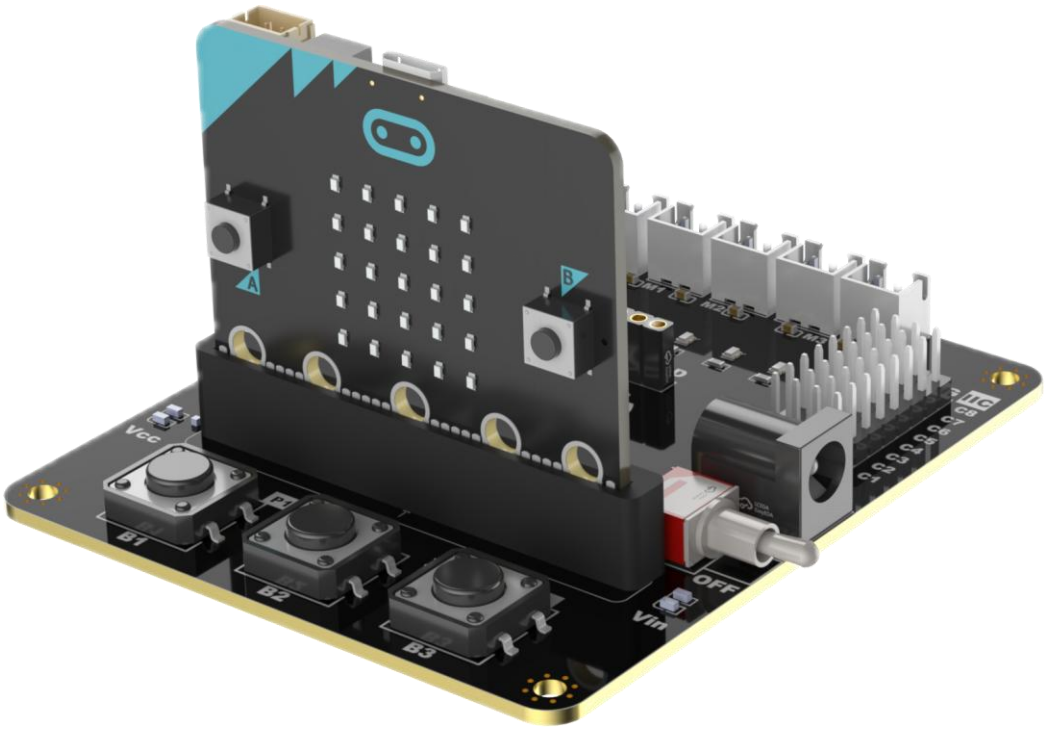
Peanut KING micro:bit Shield V2.1



Introduction

Peanut KING micro:bit Shield is an extension module used with micro:bit to enhance the usability and extensibility. The Shield can help with developing different projects easily and efficiently. Including 5 AT8236 motor drivers, providing 5 motor ports with 2A peak current per way. The extension Shield supports VCC input voltage range from 6V-24V. Providing DC power jack for power supply.

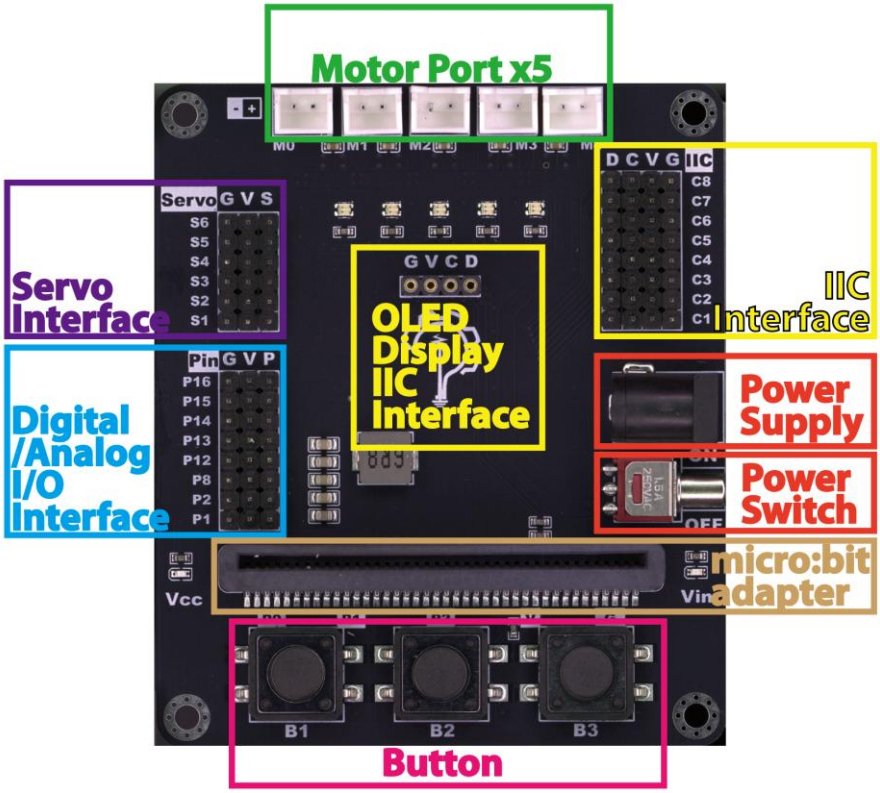
Features

Extended DuPont pins for digital/analog signal transmission with other modules
PWM motor interface : 5
micro:bit interface : 8
Servo interface : 6
I^2C interface : 8

Specification

Power supply voltage : 6 ~ 24V DC
Logic Control Voltage : 3.3V (From micro:bit)
Operating temperature: −40 ~ 85°C
Operating Humidity : < 90%
Weight : 49.8 ±1.5 g
Size : 86x73mm
5-way motor drive
Up to 2A current each way
Support PWM speed control (50Hz)

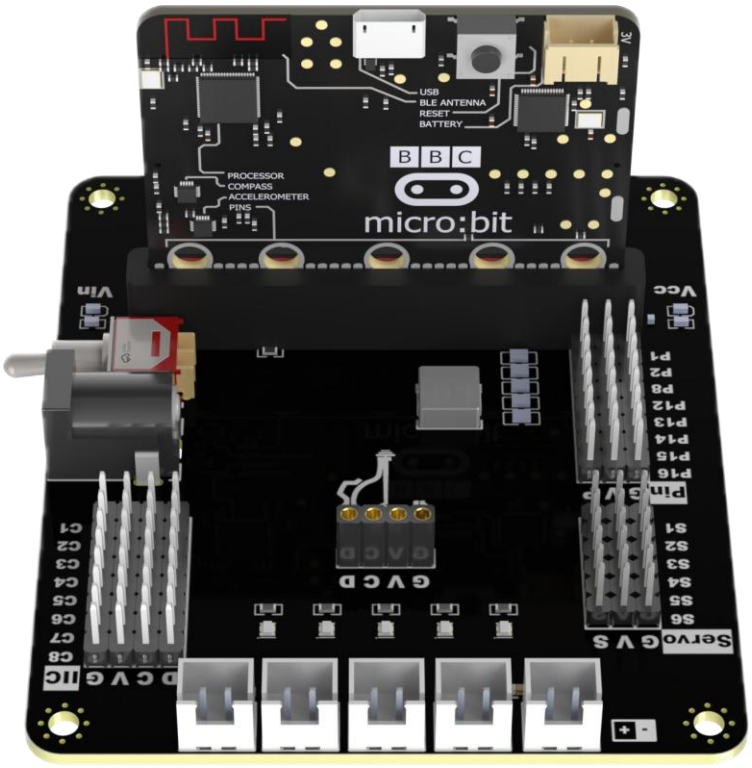
micro:bit V2.1 Sheild



Peanut KING micro:bit Shield provided 5 motor ports M0-M4 with high load driving ICs powered by Peanut KING micro: bit Power Supply with safety power switch. Moreover, 8 I^2C ports allow multiple usage of devices with same I^2C address, switching I^2C port using PKS Driver extension. 8 Servo interfaces allow the use of Servo simultaneously. IO interface connecting to micro:bit directly with respect to the marking.

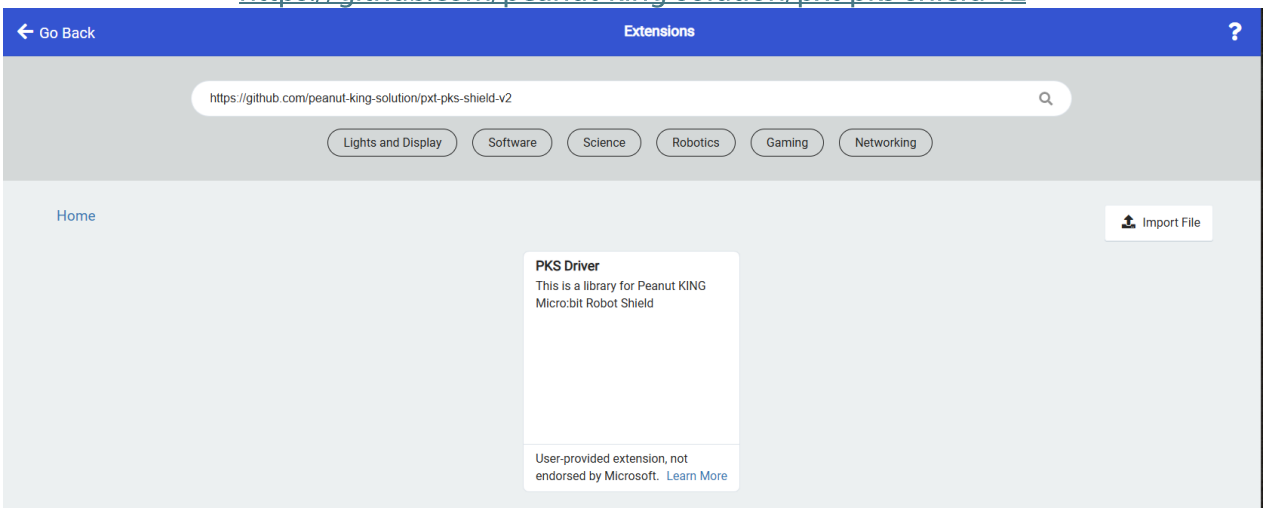
Application

Plug in micro:bit to the Peanut KING micro:bit Shield according to the marking on the Shield. The marking of the Shield and micro:bit need to be matched.



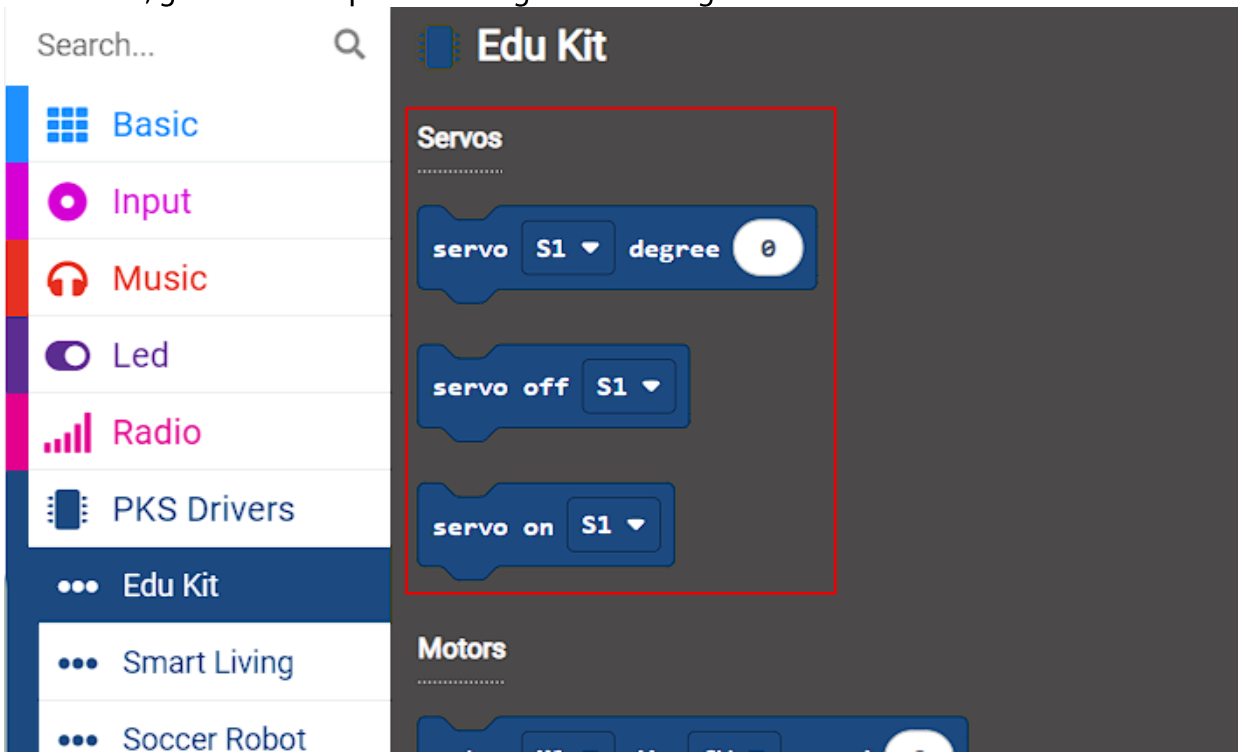
Create MakeCode Project and Install PeanutKing Shield extension with the URL provided below

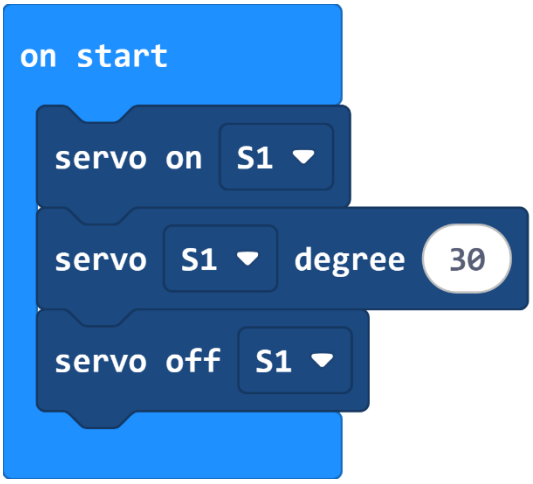
https://github.com/peanut-king-solution/pxt-pks-shield-v2



Servo:
The Peanut KING micro:bit Shield features dedicated high-precision PWM servo control ports, which is compatible with standard 9g micro-servos and high-torque mini/standard servos.

1. Connect a 9g server to the dedicated servo header (S1) on the Shield
2. In "Edu Kit", go to "Servos" part and drag the following blocks



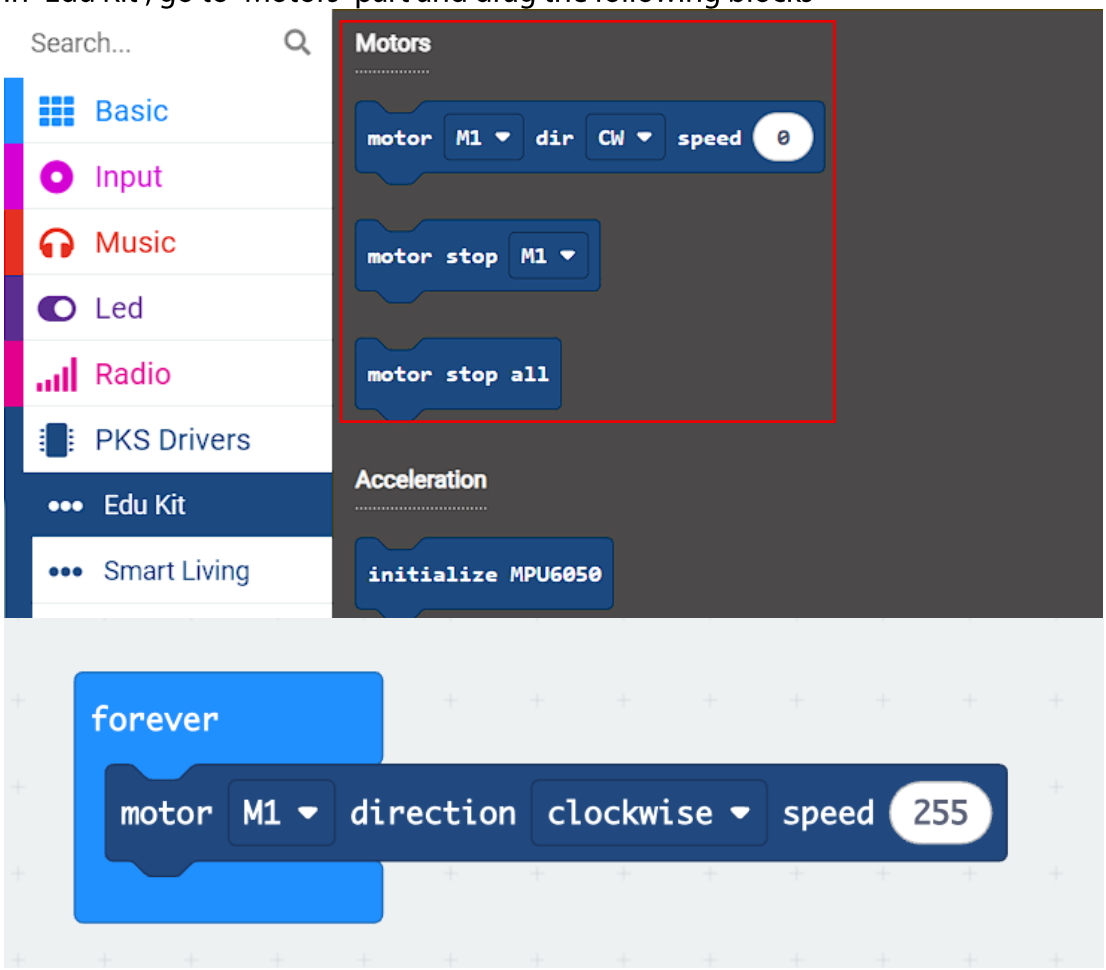


This code enable power output to the servo port using software power enable function and set the target position by writing the angular value to the servo.
Time Sequence:
Startup: Servo connected to S1 initializes and rotates to 180°.
T = 100ms: Servo connected to S1transitions to 30°.
T= 200ms: Servo connected to S1 turns back to 0°.

Motors:

Peanut KING micro:bit sheild V2.1 provided 5 high load motor port to control. Connect the DC motor leads to the corresponding XH2.54-2P motor port (M0 to M4). The rotation direction can be set by Motors Control Software block (Clockwise or Counterclockwise). Motor speed is governed by an 8-bit PWM duty cycle parameter ranging from 0 (Stop) to 255 (Maximum Speed).

1. Connect a DC motor to the XH2.54-2P motor port M1
2. In "Edu Kit", go to "Motors" part and drag the following blocks



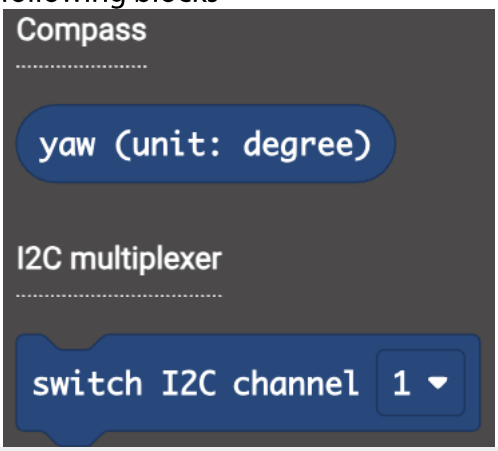
This code enable the motor connected to port M1 to rotate clockwise at full speed.

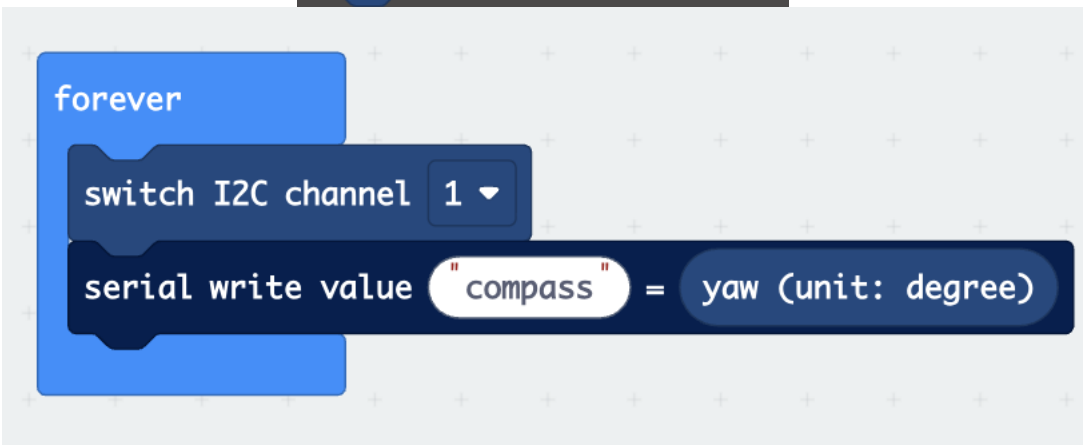
I²C Interface:

Prior to executing I²C device commands, the software must issue a channel select command to route the bus to the desired port. Placing the channel selection function before device routines ensures reliable communication and enables the use of multiple slave devices sharing identical I²C addresses without address conflicts.

Compass:
The Peanut KING Compass Module communicates with the Peanut KING micro:bit Shield via the dedicated onboard I²C bus. To interface the module, connect it to any available I²C port on the Shield and issue an I²C channel selection command prior to reading heading or magnetic field data.

1. Connect the Compass module V2.7 to the I²C interface port C1
2. In "Soccer Robot", drag the following blocks





The preceding code block switches the active I²C bus multiplexer channel to C1, establishing communication with the connected Peanut KING Compass Module. Upon channel activation, heading orientation data ranging from 0° to 359° is output via the serial interface, representing the real-time angular orientation of the device.

Ultrasound:

To interface an ultrasonic distance sensor with the Shield, connect the module to the designated Digital/Analog I/O interface using the specialized custom DuPont cable from Peanut KING Solution. This dedicated cable harness ensures correct pinout mapping and signal integrity for the Trigger and Echo lines, providing reliable power supply and accurate non-contact distance measurement.

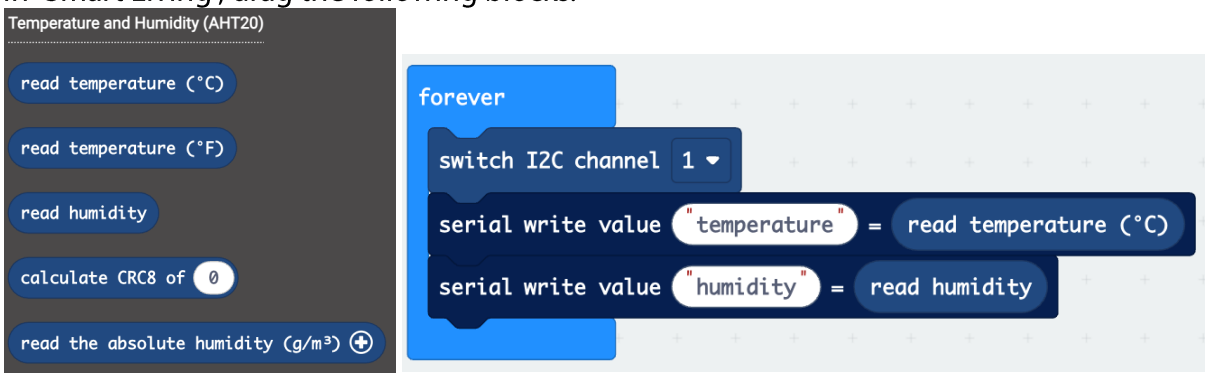
1. Connect the ultrasonic module to the dedicated port at Digital/Analog I/O interface refering to the teaching notes.
2. In "Maze Car", drag the following blocks.

The preceding code block output the real-time distance between the sensor and obstacle via serial interface.

Temperature and Humidity Sensor:

The AHT20 digital temperature and humidity sensor interfaces with the Shield via the dedicated I^2C bus. Connect the AHT20 module directly to the Shield's I^2C interface port. Prior to reading environmental data (relative humidity and ambient temperature), the software must select the active I^2C channel corresponding to the connected port to ensure proper signal routing and communication with the sensor.

1. Connect the AHT20 module to the dedicated port at I^2C interface refering to the teaching notes.
2. In "Smart Living", drag the following blocks.

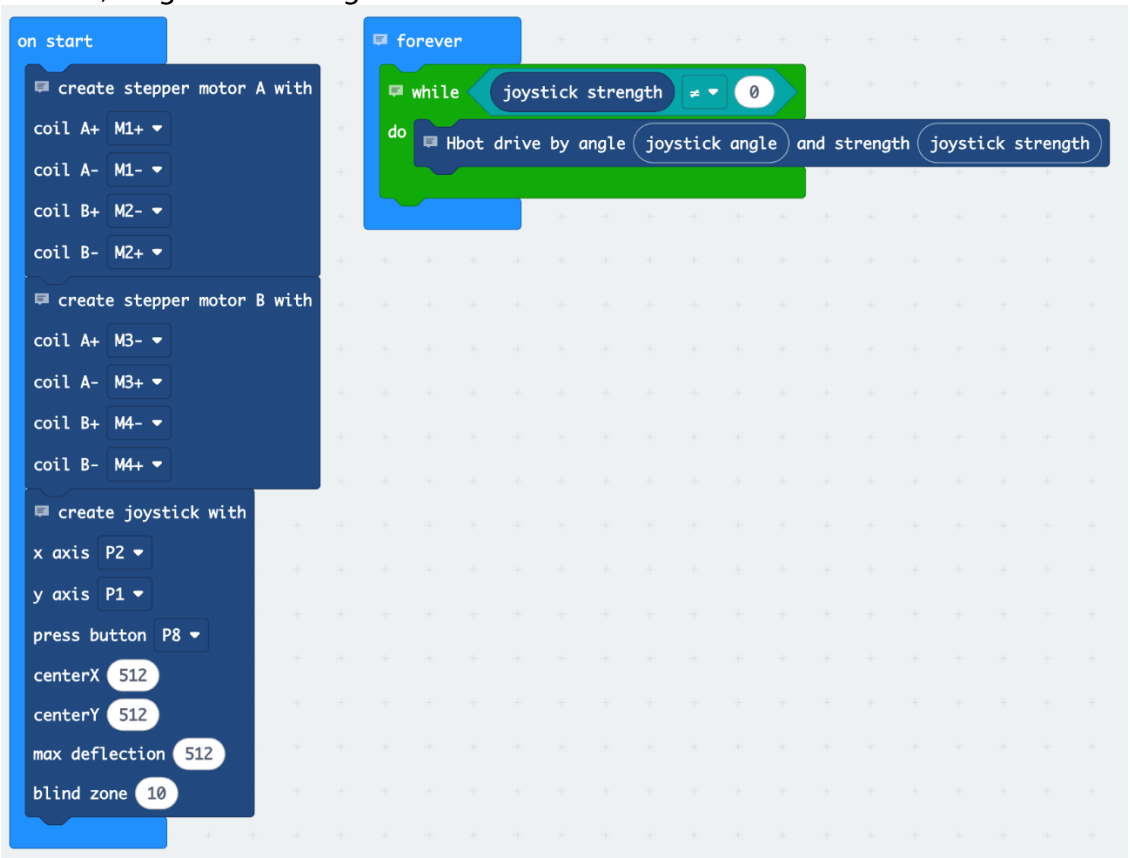


The preceding code block switches the active I^2C bus multiplexer channel to C1, establishing communication with the connected AHT20 module. Upon channel activitation, temperature and humidity of the ambient are output via the serial interface.

Gotcha Machine:

The system features integrated dual stepper motor control configured in an H-Bot kinematic matrix. This architecture allows proportional X-Y planar axis movement driven by analog joystick inputs, ideal for claw mechanism positioning in Gachapon or arcade-style applications.

1. Refer the the Gotacha machine teaching notes and assemble the machine.
2. In "Gotcha", drag the following blocks.



The application initializes by defining the pin mapping for Stepper Motor A (using channels M1 and M2) and Stepper Motor B (using channels M3 and M4). Next, the joystick routine registers analog inputs P2 (X-axis) and P1 (Y-axis), alongside digital pin P8 (button switch), establishing baseline centering thresholds at 512 with a noise tolerance deadband of 10.

Within the background execution loop, the firmware continuously polls for joystick displacement (JoystickStrength() != 0). Upon detecting deflection beyond the deadband threshold, the system computes the vector angle and magnitude, passing these values to the HBotMoveByJoystick() function to calculate required step frequencies for both motors to perform precise planar X-Y movement.

Important Information and Disclaimer

The information in this statement reflects Peanut King Solution Limited's knowledge and belief as of the date it is provided. Peanut King Solution Limited relies on information from third parties and does not guarantee its accuracy. Efforts are ongoing to better integrate third-party information. Peanut King Solution Limited has taken and continues to take reasonable steps to provide accurate and representative information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. Certain information is considered proprietary by Peanut King Solution Limited and its suppliers, so CAS numbers and other limited information may not be available for release.

NOTICE

Peanut King Solution Limited reserves the right to make changes to the products in this document to improve performance or for any other reason, or to discontinue them without notice. Customers should contact Peanut King Solution Limited for the latest product information before placing orders or designing Peanut King Solution Limited products into systems. Peanut King Solution Limited assumes no responsibility for the use of any products or circuits described in this document or customer product design, does not convey any license under any patent or other right, and does not guarantee that the circuits are free of patent infringement. Peanut King Solution Limited makes no claim regarding the suitability of its products for any particular purpose and assumes no liability arising from the use of any product or circuit, specifically disclaiming any and all liability, including consequential or incidental damages.

PEANUT KING SOLUTION LIMITED PRODUCTS ARE NOT DESIGNED OR INTENDED FOR USE IN CRITICAL APPLICATIONS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY OR DEATH. THE USE OF PEANUT KING SOLUTION LIMITED PRODUCTS IN LIFE SUPPORT SYSTEMS IS NOT AUTHORIZED.