



# MicroBlocks on Zephyr

By: Ayush Singh



# Today's Talk

- MicroBlocks Overview
- Using MicroBlocks on Zephyr board (BeagleConnect Freedom)
- Porting MicroBlocks to more Zephyr supported boards
- Future



# About Me

Github: @Ayush1325

- Student at Indian Institute of Technology (ISM), Dhanbad
- Embedded System Developer for BeagleBoard.org
- Google Summer of Code 2022 and 2023 contributor



# About BeagleBoard.org

- The BeagleBoard.org Foundation is a Michigan, USA-based 501(c)(3) non-profit corporation existing to provide education in and collaboration around the design and use of open-source software and hardware in embedded computing.
- Has a wide range of embedded boards with
  - Linux support: BeagleY-AI, BeaglePlay, BeagleV-Ahead, etc
  - Zephyr support: BeagleV-Fire, BeagleConnect Freedom, etc
- All Beagles are open-hardware with documentation on the design, including the materials to modify the designs from source using appropriate design tools.

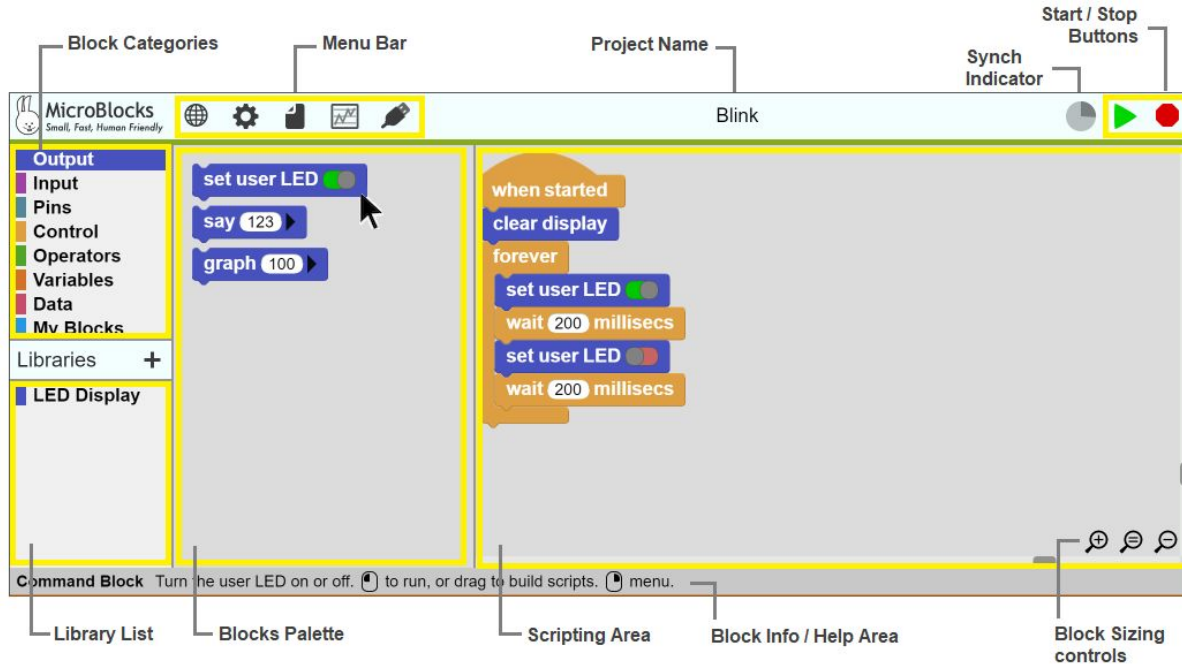




# About MicroBlocks

- What is MicroBlocks?
  - It is a free, Scratch-like blocks programming language for learning physical computing with educational microcontroller boards such as the micro:bit, Adafruit Circuit Playground Express, and many others.
- What makes MicroBlocks different?
  - While there are other blocks languages for microcontrollers, what really sets MicroBlocks apart is its combination of live programming and autonomous operation. Other blocks languages support one or the other of those features but not both.

# MicroBlocks IDE





# MicroBlocks Capabilities

- Program Persistence
- Peripherals: GPIO, ADC, PWM, SPI, I2C, UART, etc
- Networking: BLE
- Custom Libraries





# MicroBlocks Internals

- A Virtual Machine (VM) that interprets and executes it's own lisp like language generated from blocks.
- Implemented using Arduino APIs.
- PlatformIO build system.
- Based on GP Blocks.



## About MicroBlocks for Zephyr

- A freestanding Zephyr Application: <https://openbeagle.org/beagleboard/microblocks>
- Pulls upstream Zephyr, MicroBlocks VM, Arduino module for Zephyr and upstream Arduino Core API using west.
- Contains a small main function and a logging backend to divert logs to MicroBlocks IDE.
- Contains west manifest to build dev, pilot and stable branches of MicroBlocks VM.
- Any MicroBlocks specific board overlay



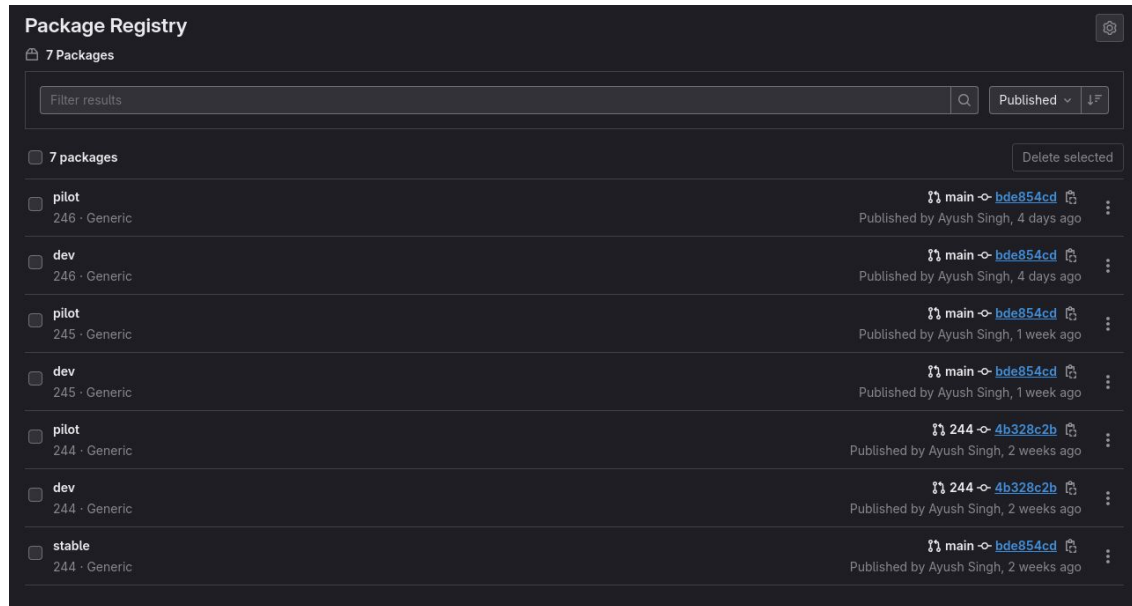
## Why use Arduino Module for Zephyr

- MicroBlocks is currently implemented using Arduino APIs.
- BeagleConnect Freedom already has good Zephyr support.
- Due to MicroBlocks being a small team, they cannot maintain another platform. Thus anything too Zephyr centric will need to be maintained out of tree.
- Since we need to maintain Zephyr centric stuff out of tree, it is better to keep the footprint minimal to reduce maintenance burden.

---

# Running MicroBlocks on BeagleConnect Freedom

# Prebuilt Images



The screenshot displays the OpenBeagle Package Registry interface. At the top, it says "Package Registry" and "7 Packages". Below this is a search bar with the text "Filter results" and a search icon. To the right of the search bar are two buttons: "Published" with a dropdown arrow and a "Sort" icon. Below the search bar, there is a "7 packages" summary and a "Delete selected" button. The main content is a list of 7 packages, each with a checkbox, a name, a version, a description, and a "Published by" field with a timestamp. The packages are:

Package Name	Version	Description	Published by	Time Ago
pilot	246	Generic	Ayush Singh	4 days ago
dev	246	Generic	Ayush Singh	4 days ago
pilot	245	Generic	Ayush Singh	1 week ago
dev	245	Generic	Ayush Singh	1 week ago
pilot	244	Generic	Ayush Singh	2 weeks ago
dev	244	Generic	Ayush Singh	2 weeks ago
stable	244	Generic	Ayush Singh	2 weeks ago

<https://openbeagle.org/beagleboard/microblocks/-/packages>



# Build MicroBlocks for Zephyr

- **Create a workspace folder:**

- `mkdir microblocks-workspace && cd microblocks-workspace`

- **Setup virtualenv**

- `python -m venv .venv && source .venv/bin/activate`

- `pip install west`

- **Setup Zephyr app**

- `west init -m https://openbeagle.org/beagleboard/microblocks.git --mf west_dev.yml .`

- `west update`



## Build MicroBlocks for Zephyr (Continued)

- **Setup Arduino module**

- `ln -srf modules/lib/ArduinoCore-API/api modules/lib/Arduino-Zephyr-API/cores/arduino/.`

- **Install python deps**

- `pip install -r zephyr/scripts/requirements-base.txt`

- **Build**

- `west build -b beagleconnect_freedom microblocks -p`

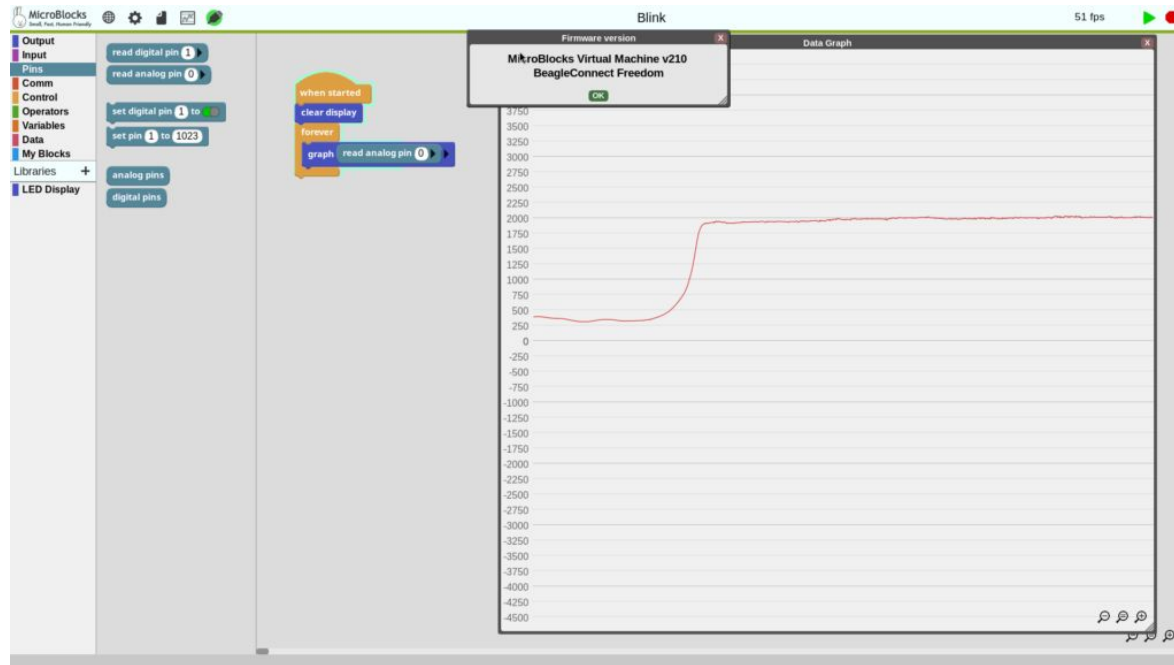


# Flash BeagleConnect Freedom

- **Install `cc1352-flasher`**
  - `pip install cc1352-flasher`
- **Flash image**
  - `cc1352_flasher --bcf ~/zephyr.bin -p '/dev/ttyACM0'`



# MicroBlocks running on BeagleConnect Freedom





# Porting to new boards

- Add board support to Arduino Core API for Zephyr
- Add board support to upstream MicroBlocks
- Add board-specific overlay to MicroBlocks Application

---

# Arduino Core API for Zephyr



## Define pins

- The default i2c pins
- Needed by MicroBlocks to do i2c initialization

```
#define PIN_WIRE_SCL D12  
  
#define PIN_WIRE_SDA D11
```



# Upstream MicroBlocks



# Add board pin details to MicroBlocks

- Add to vm/ioPrims.cpp

```
#elif defined(CONFIG_BOARD_BEAGLECONNECT_FREEDOM)

    #define BOARD_TYPE "BeagleConnect Freedom"

    #define DIGITAL_PINS 24

    #define ANALOG_PINS 6

    #define TOTAL_PINS 24

    static const int analogPin[] = {A0, A1, A2, A3, A4, A5};

    #define PIN_LED LED_BUILTIN

    #define DEFAULT_TONE_PIN D6

    #define PIN_BUTTON_A D7

    #define PIN_BUTTON_B D8
```



## Add Memory

- Amount of memory available for vm stack
- vm/mem.c

```
#elif defined(BOARD)
```

```
#define OBJSTORE_BYTES 40000
```

---

# MicroBlocks Zephyr Application





## Persist Partition

- Add board overlay for MicroBlocks persistence partition.
- Note: The partition should allow address access, i.e. be memory mapped
- Eg: boards/beagleconnect\_freedom.overlay

```
&flash0 {
    partitions {
        /* Allocate 568 KiB for persistence */
        persist_partition: partition@1F400 {
            reg = <0x0001F400 DT_SIZE_K(568)>;
        };
    };
};
```

**Your board is now supported**





# Next Steps

- **Add MicroBlocks Zephyr app upstream**
  - MicroBlocks currently uses PlatformIO for building.
- **Support More Zephyr Boards**
  - Expand MicroBlocks to more Zephyr boards
- **Servo support**
  - Currently blank implementations for servo on Zephyr
- **Adapt to Arduino on Zephyr**
  - With Arduino's announcement to switch to Zephyr, things will probably change a lot going forward.



## References

- MicroBlocks: <https://microblocks.fun>
- Arduino Core API module for Zephyr: <https://github.com/zephyrproject-rtos/gsoc-2022-arduino-core>
- GP Blocks: <https://gpblocks.org/>
- Zephyr: <https://zephyrproject.org/>
- MicroBlocks application: <https://openbeagle.org/ayush1325/microblocks>

**Thank You**

