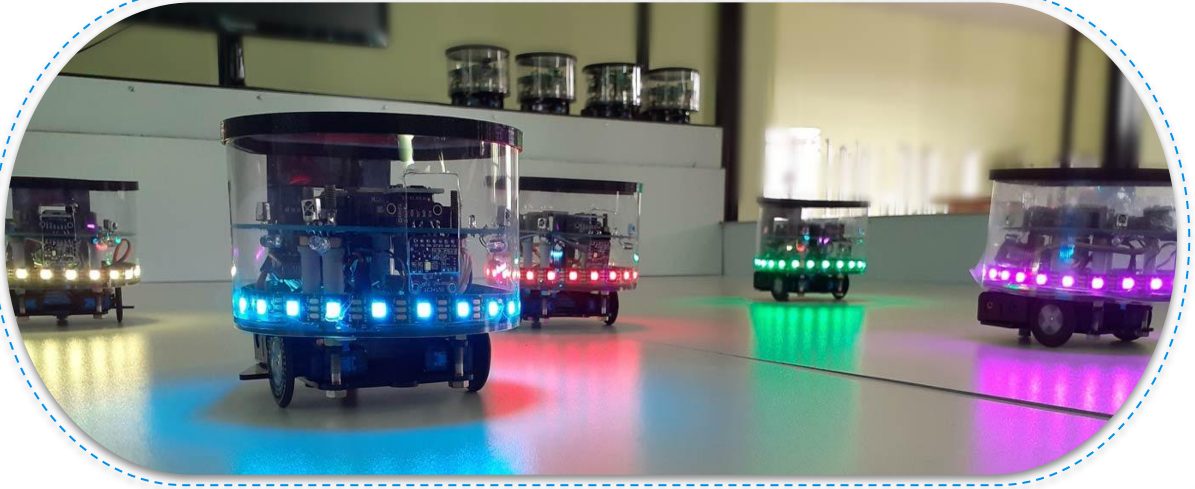
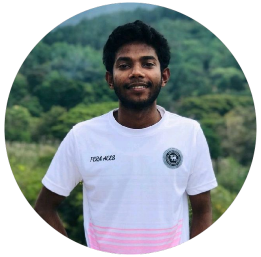
Swarm Intelligence Programming Framework



By Group 03

Members


Ekanayake S.M.

E/16/094

Madushanka H.M.K.

E/16/221

Perera A.L.H.E.

E/16/275

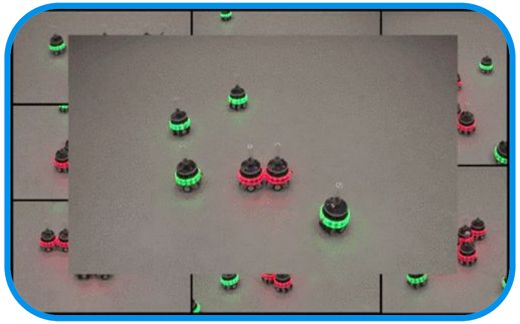
Supervisors

Prof. Roshan Ragel

Dr. Isuru Nawinne

Dr. Mahanama Wickramasinghe

Dr. Sithumini Ekanayake



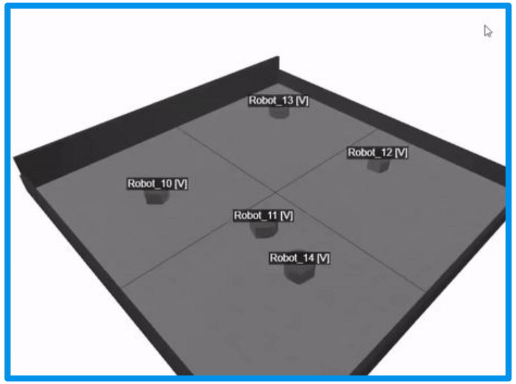
Swarm Robotics

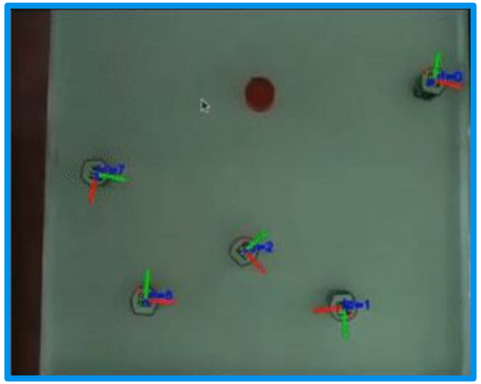
1

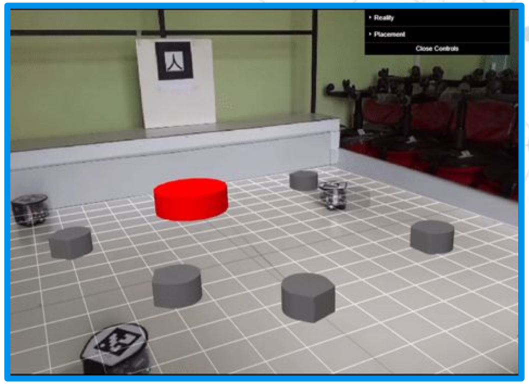
Introduction

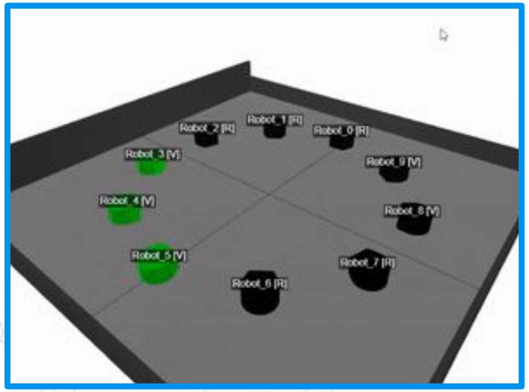
Introduction to PeraSwarm Project

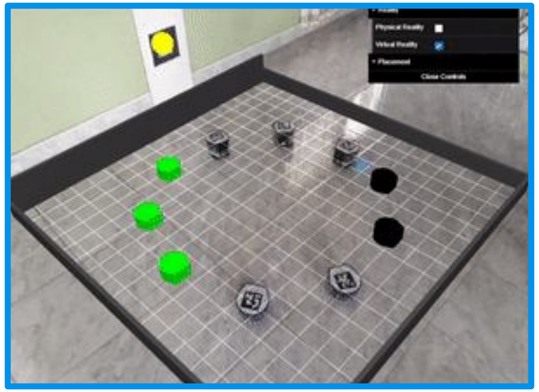
PeraSwarm

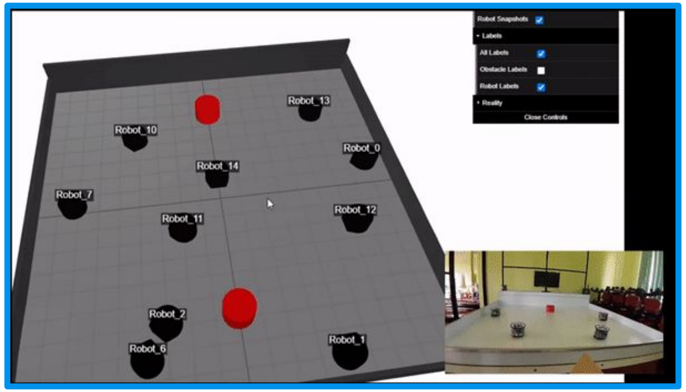

Robot_13 [V]
Robot_12 [V]
Robot_10 [V]
Robot_11 [V]
Robot_14 [V]




Reality
Placement
Close Controls


Robot_2 [R]
Robot_1 [R]
Robot_0 [R]
Robot_3 [V]
Robot_9 [V]
Robot_4 [V]
Robot_8 [V]
Robot_5 [V]
Robot_7 [R]
Robot_6 [R]


Physical Reality
Virtual Reality
Placement
Close Controls


Robot Snapshots
Labels
All Labels
Obstacle Labels
Robot Labels
Reality
Close Controls
Robot_13
Robot_10
Robot_0
Robot_14
Robot_7
Robot_11
Robot_12
Robot_2
Robot_6
Robot_1

2

Problems

Issues that are going to solve

Problems

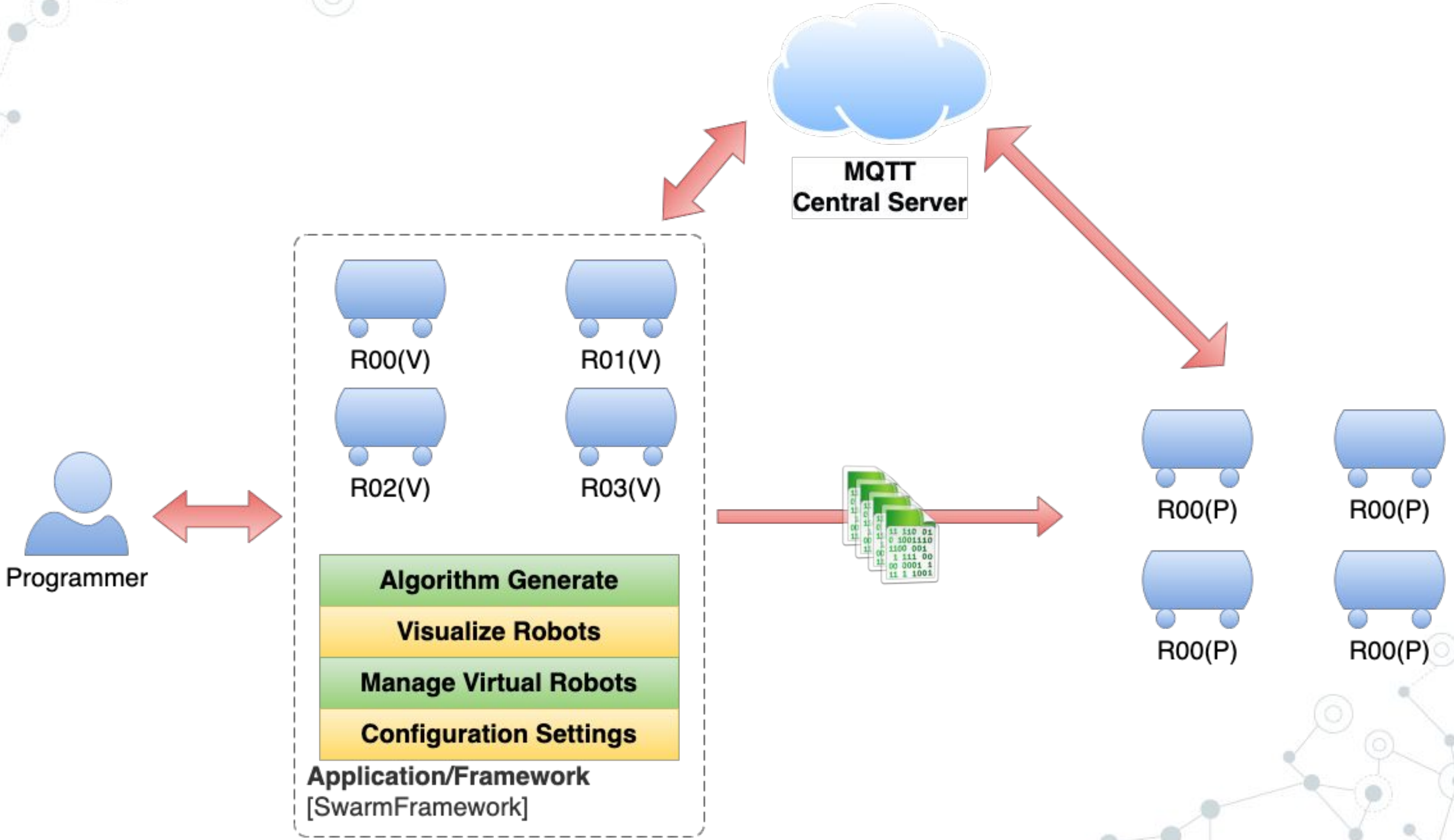
- No framework that supports different physical and virtual robots.
- Inability to programme multiple robots over the air.
- Swarm programmers must be experts in programming.

3

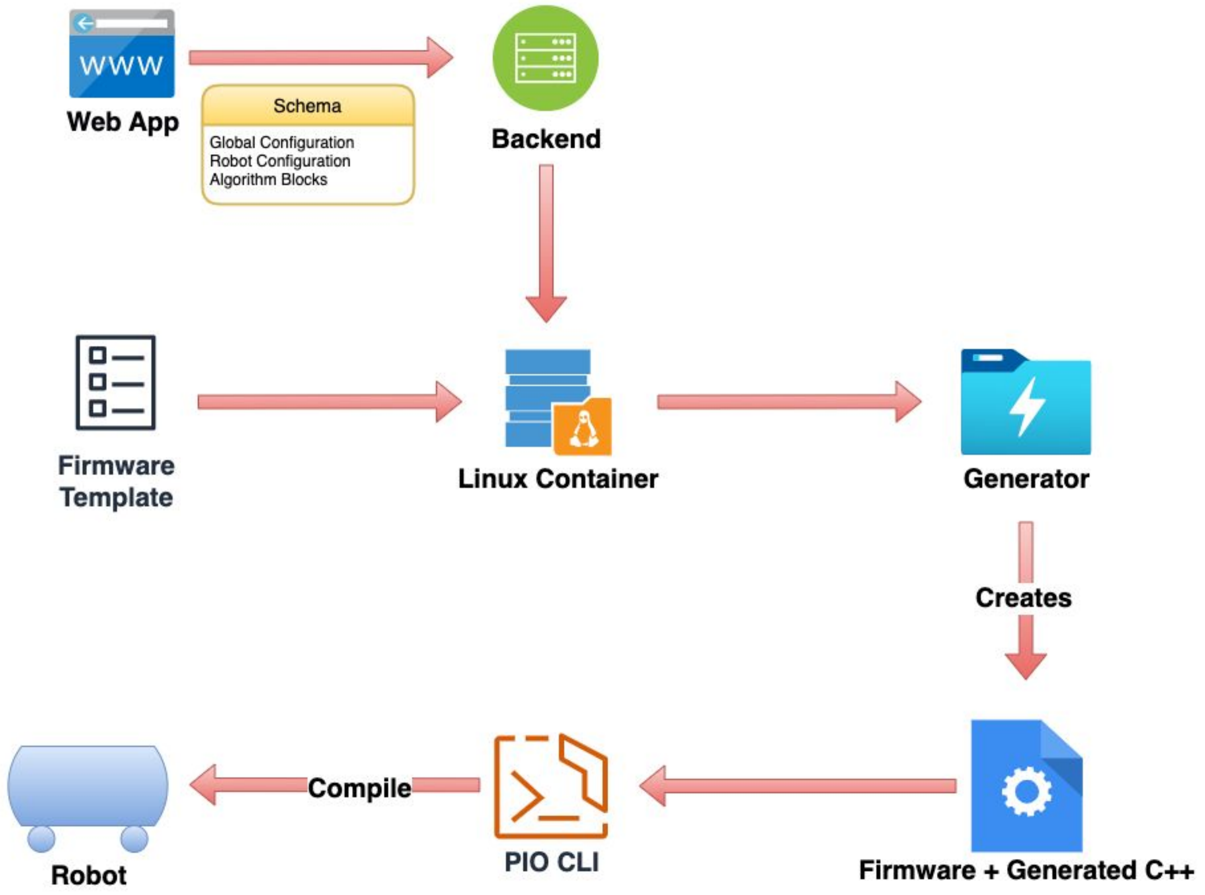
Solution

Solution Architecture

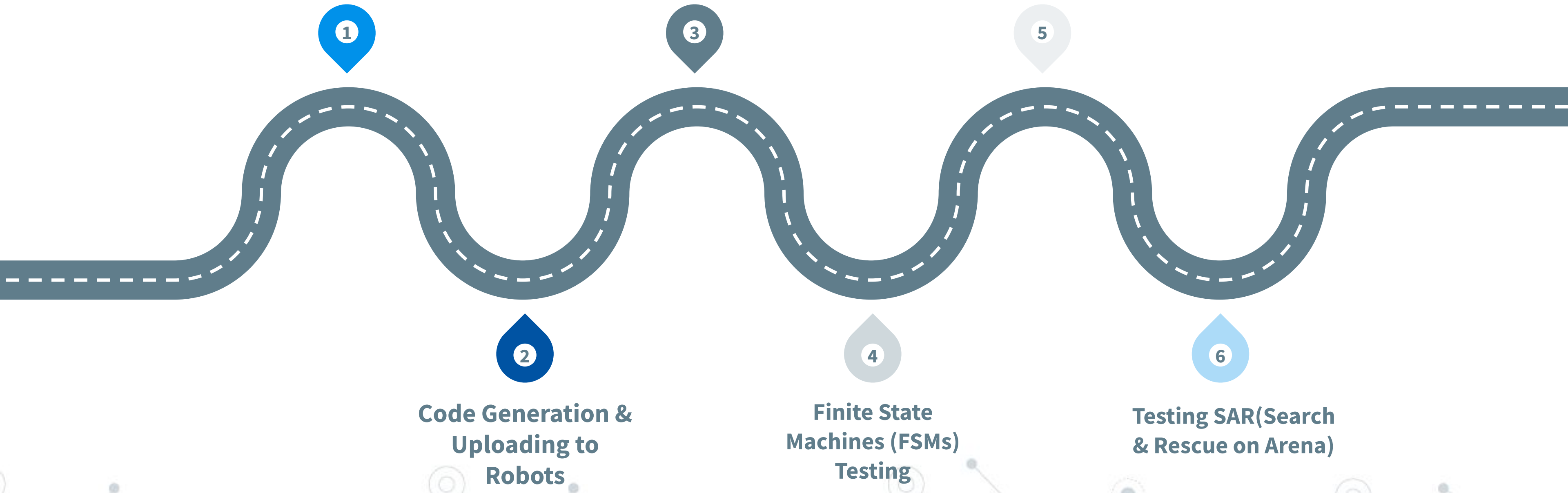
High level overview



Software Architecture



Roadmap



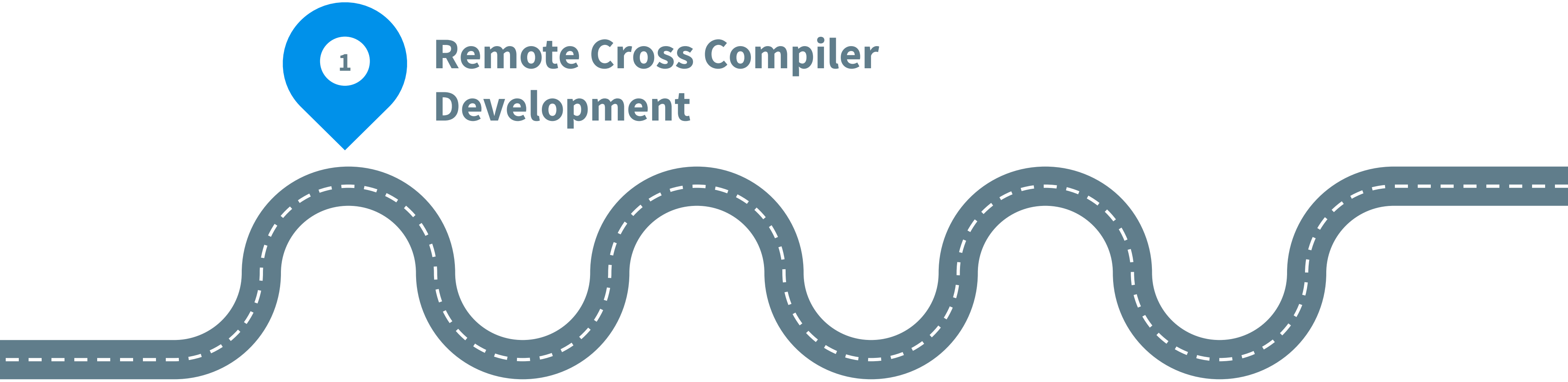
Remote Cross Compiler Development
1
Code Generation & Uploading to Robots
2
Algorithms Development & Testing
3
Finite State Machines (FSMs) Testing
4
Aggregator FSM Testing
5
Testing SAR(Search & Rescue on Arena)
6

7

Progress

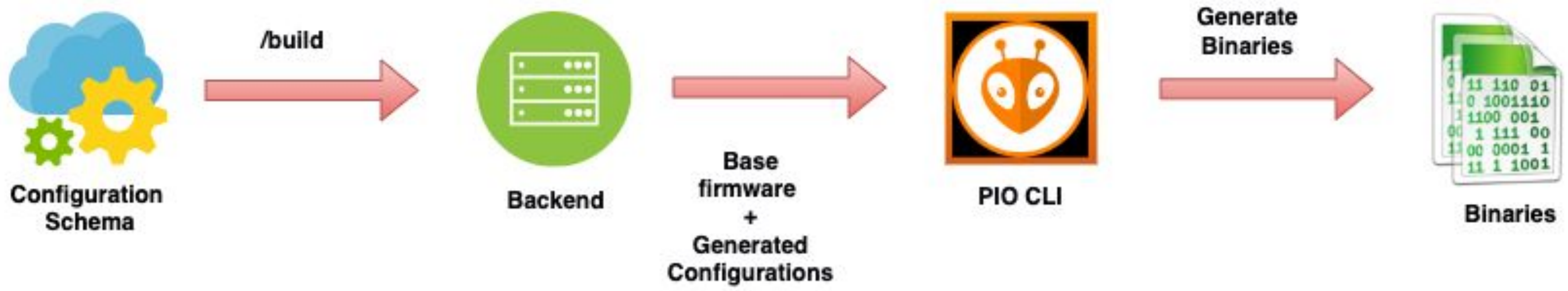
Current Progress Status

Roadmap

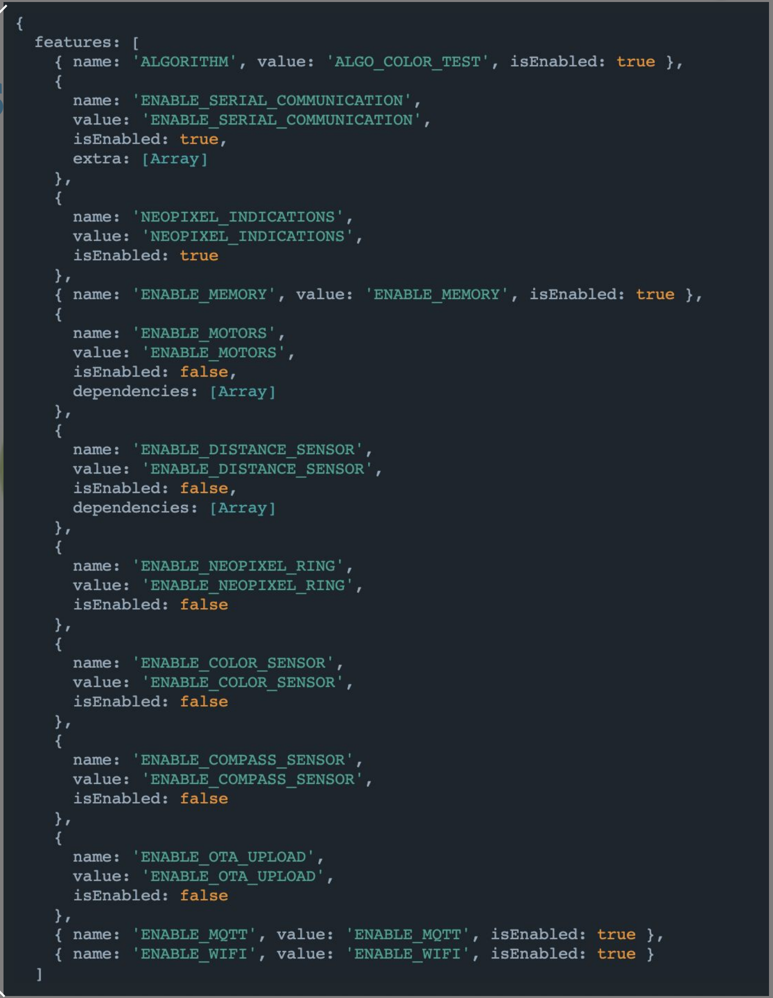


Remote Cross Compiler Development

Remote Cross Compiler Development



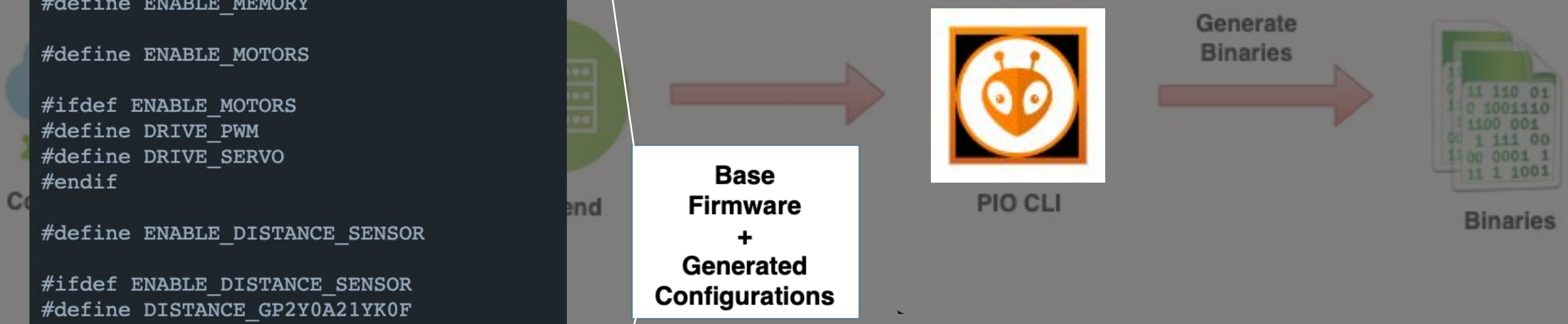
Remote Cross



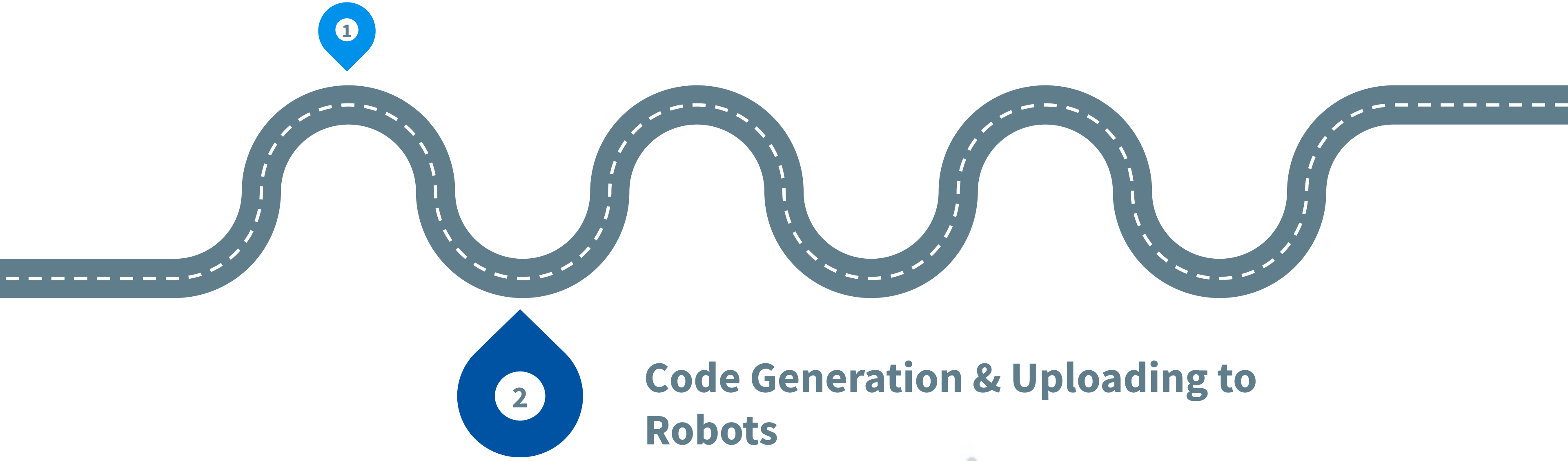
```
{
  features: [
    { name: 'ALGORITHM', value: 'ALGO_COLOR_TEST', isEnabled: true },
    {
      name: 'ENABLE_SERIAL_COMMUNICATION',
      value: 'ENABLE_SERIAL_COMMUNICATION',
      isEnabled: true,
      extra: [Array]
    },
    {
      name: 'NEOPIXEL_INDICATIONS',
      value: 'NEOPIXEL_INDICATIONS',
      isEnabled: true
    },
    { name: 'ENABLE_MEMORY', value: 'ENABLE_MEMORY', isEnabled: true },
    {
      name: 'ENABLE_MOTORS',
      value: 'ENABLE_MOTORS',
      isEnabled: false,
      dependencies: [Array]
    },
    {
      name: 'ENABLE_DISTANCE_SENSOR',
      value: 'ENABLE_DISTANCE_SENSOR',
      isEnabled: false,
      dependencies: [Array]
    },
    {
      name: 'ENABLE_NEOPIXEL_RING',
      value: 'ENABLE_NEOPIXEL_RING',
      isEnabled: false
    },
    {
      name: 'ENABLE_COLOR_SENSOR',
      value: 'ENABLE_COLOR_SENSOR',
      isEnabled: false
    },
    {
      name: 'ENABLE_COMPASS_SENSOR',
      value: 'ENABLE_COMPASS_SENSOR',
      isEnabled: false
    },
    {
      name: 'ENABLE_OTA_UPLOAD',
      value: 'ENABLE_OTA_UPLOAD',
      isEnabled: false
    },
    { name: 'ENABLE_MQTT', value: 'ENABLE_MQTT', isEnabled: true },
    { name: 'ENABLE_WIFI', value: 'ENABLE_WIFI', isEnabled: true }
  ]
```

mpiler Development

```
    #pragma once
    /*
      This is an auto-generated file.
    */

#define ALGO_COLOR_TEST

#define ENABLE_SERIAL_COMMUNICATION1

#define NEOPIXEL_INDICATIONS

#define ENABLE_MEMORY

#define ENABLE_MOTORS

#ifdef ENABLE_MOTORS
#define DRIVE_PWM
#define DRIVE_SERVO
#endif

#define ENABLE_DISTANCE_SENSOR

#ifdef ENABLE_DISTANCE_SENSOR
#define DISTANCE_GP2Y0A21YK0F
#endif

#define ENABLE_NEOPIXEL_RING

#define ENABLE_MQTT

#define ENABLE_WIFI

/* ------- End of file ------- */
```

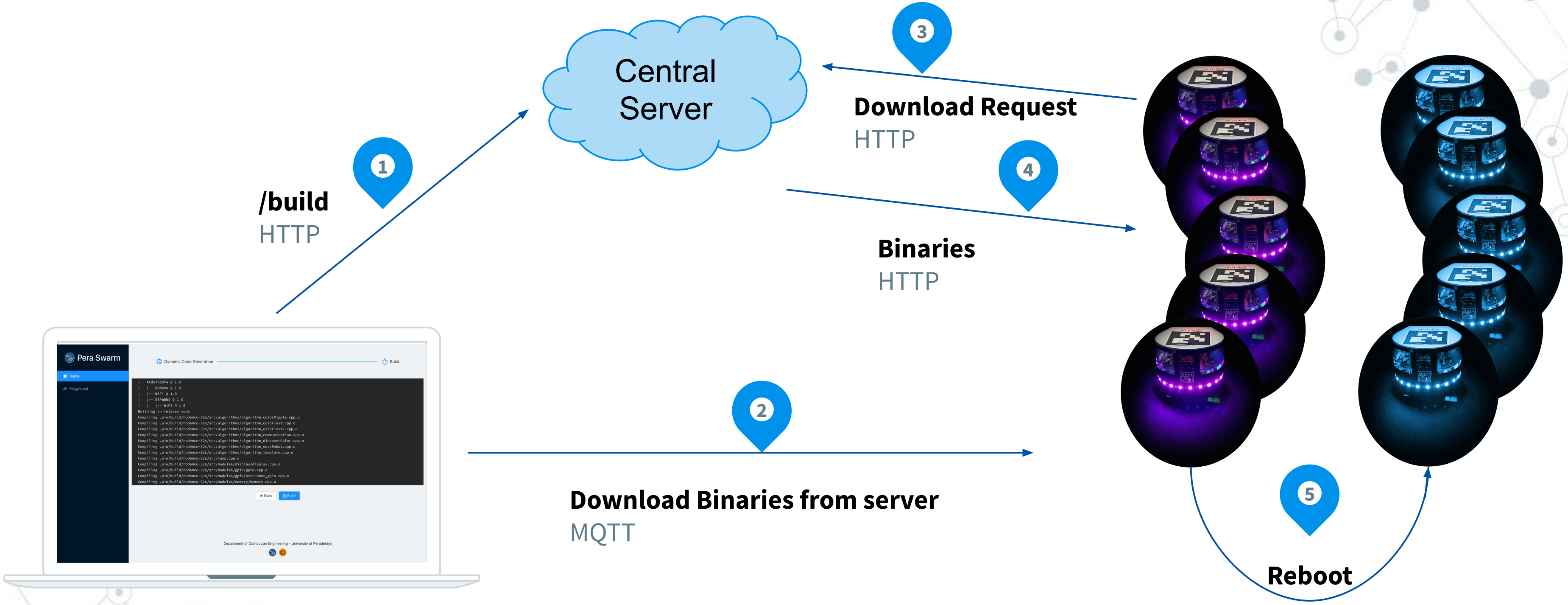


Roadmap



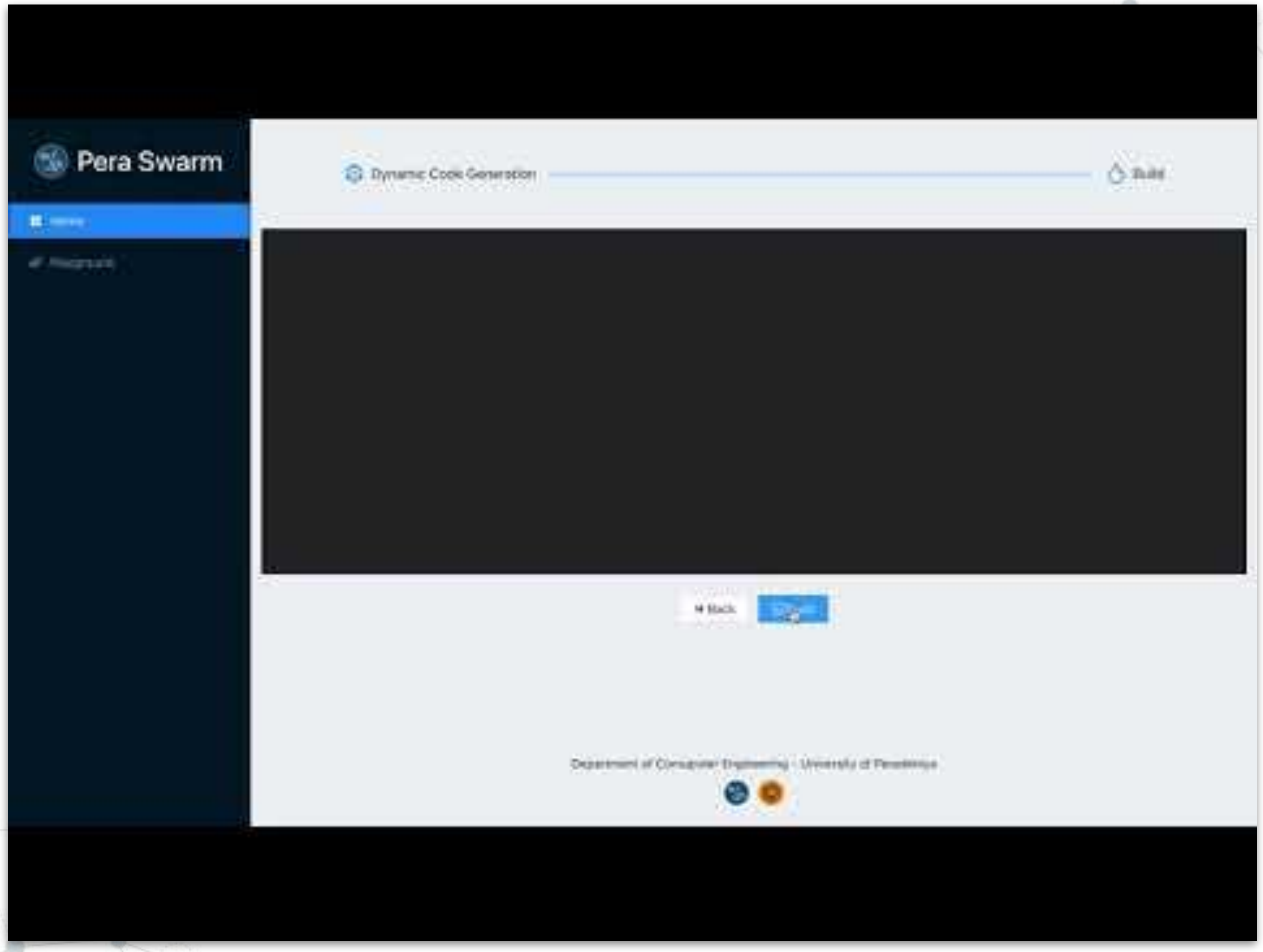
Code Generation & Uploading to Robots

OTA (Over The Air) Upload

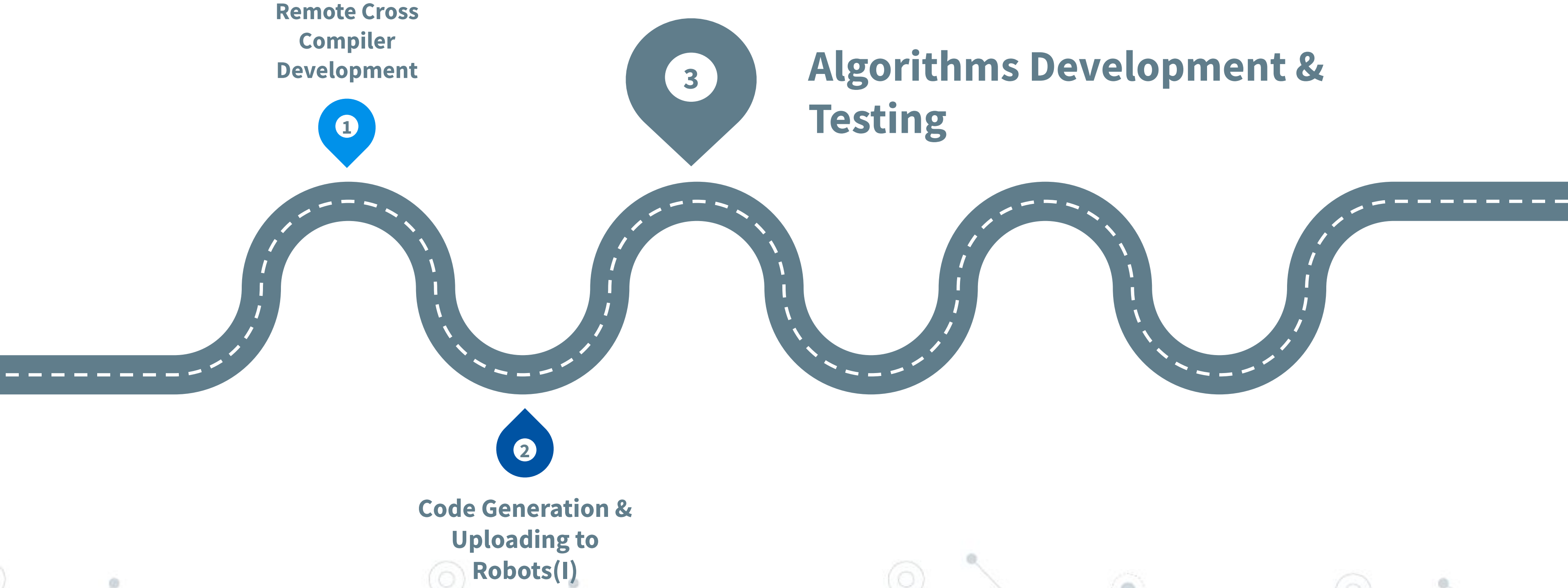


```
|-- ArduinoOTA @ 1.0
|   |-- Update @ 1.0
|   |-- WiFi @ 1.0
|   |-- ESPmDNS @ 1.0
|   |   |-- WiFi @ 1.0
Building in release mode
Compiling .pio/build/nodemcu-32s/src/algorithms/algorithm_colorRipple.cpp.o
Compiling .pio/build/nodemcu-32s/src/algorithms/algorithm_colorTest.cpp.o
Compiling .pio/build/nodemcu-32s/src/algorithms/algorithm_colorTest2.cpp.o
Compiling .pio/build/nodemcu-32s/src/algorithms/algorithm_communication.cpp.o
Compiling .pio/build/nodemcu-32s/src/algorithms/algorithm_discoverColor.cpp.o
Compiling .pio/build/nodemcu-32s/src/algorithms/algorithm_moveRobot.cpp.o
Compiling .pio/build/nodemcu-32s/src/algorithms/algorithm_template.cpp.o
Compiling .pio/build/nodemcu-32s/src/loop.cpp.o
Compiling .pio/build/nodemcu-32s/src/modules/display/display.cpp.o
Compiling .pio/build/nodemcu-32s/src/modules/gpio/gpio.cpp.o
Compiling .pio/build/nodemcu-32s/src/modules/gpio/src/robot_gpio.cpp.o
Compiling .pio/build/nodemcu-32s/src/modules/memory/memory.cpp.o
```

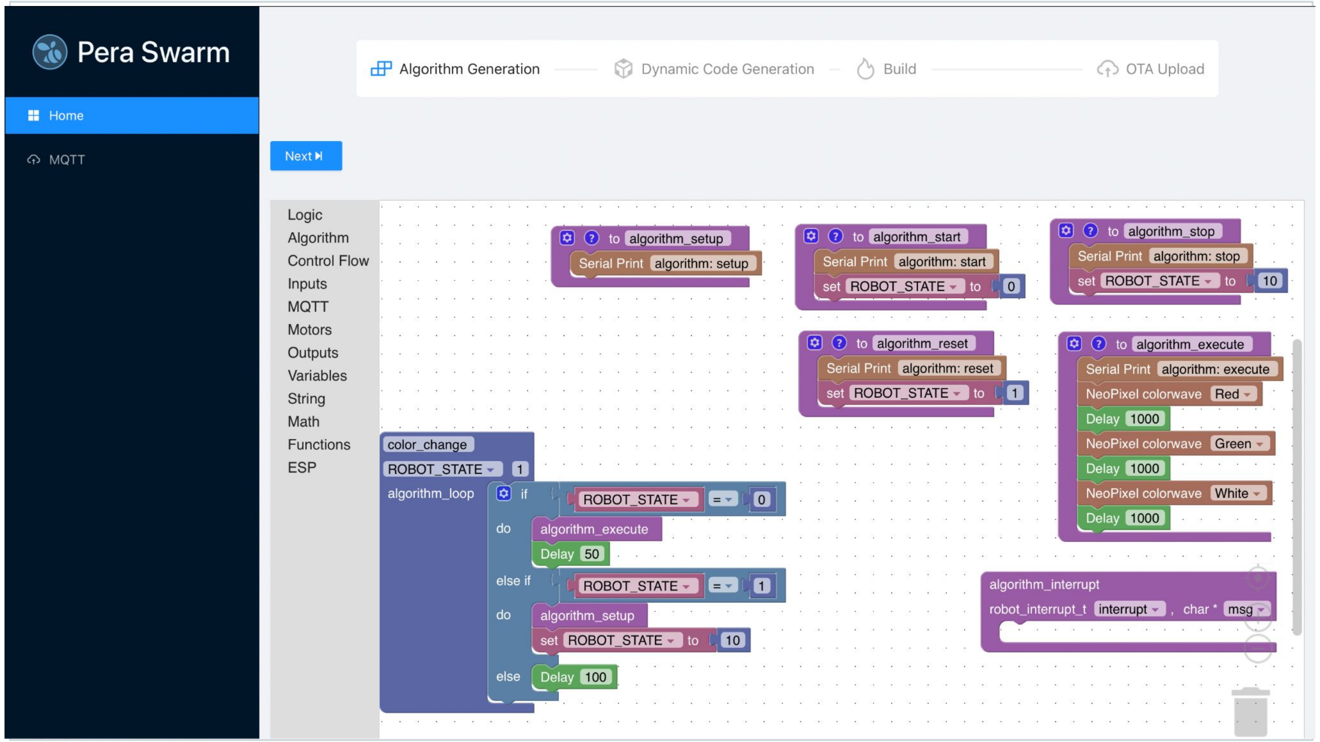
Code Generation & Uploading to Robots (I)


Pera Swarm
Home
Playground
Dynamic Code Generation
Build
Back
Department of Computer Engineering · University of Peradeniya

Roadmap


Remote Cross Compiler Development
1
3
Algorithms Development & Testing
2
Code Generation & Uploading to Robots(I)

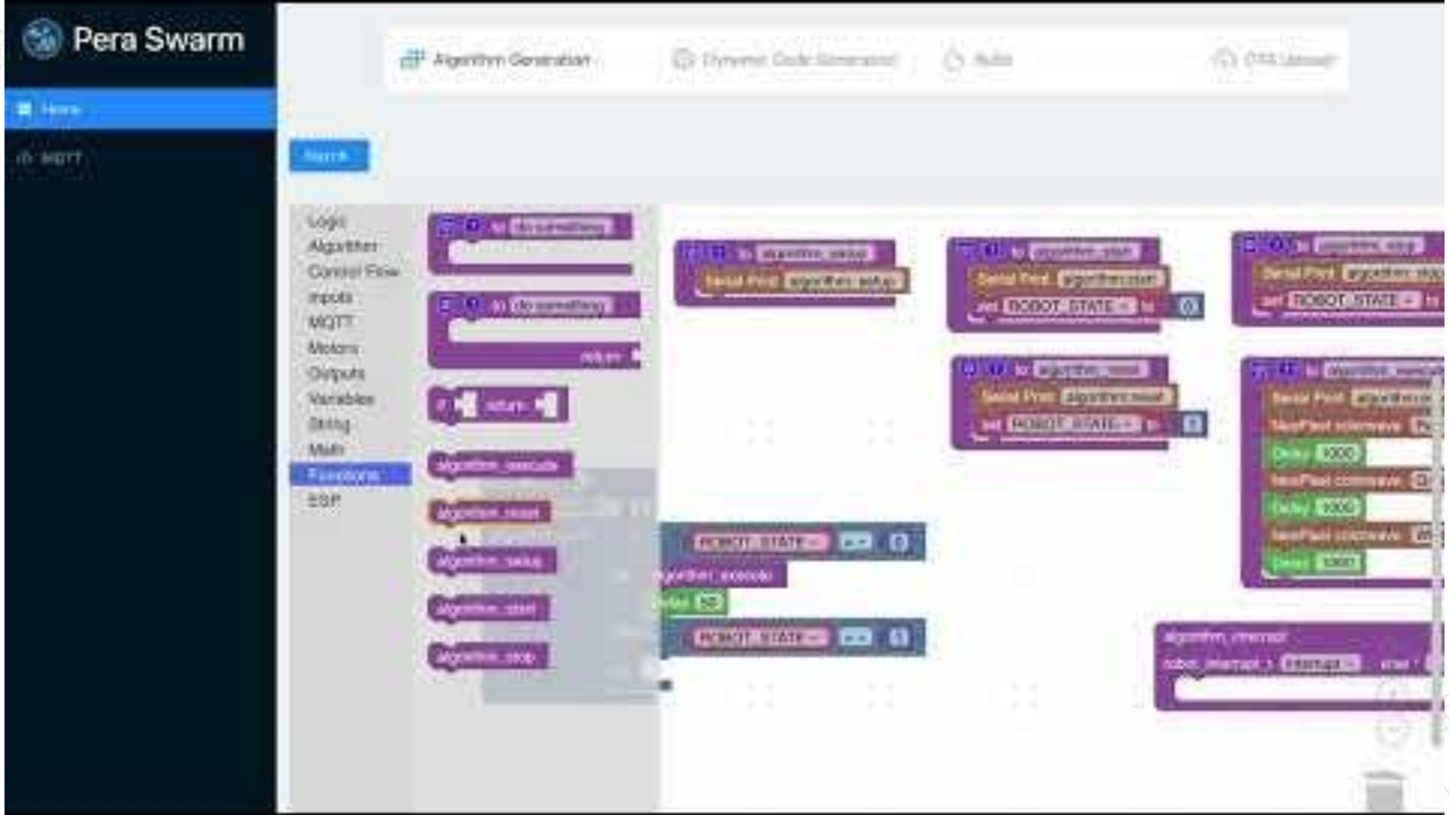
Algorithms Development & Testing



```
to algorithm_setup
  Serial Print algorithm: setup

to algorithm_start
  Serial Print algorithm: start
  set ROBOT_STATE to 0

to algorithm_stop
  Serial Print algorithm: stop
  set ROBOT_STATE to 10

to algorithm_reset
  Serial Print algorithm: reset
  set ROBOT_STATE to 1

to algorithm_execute
  Serial Print algorithm: execute
  NeoPixel colorwave Red
  Delay 1000
  NeoPixel colorwave Green
  Delay 1000
  NeoPixel colorwave White
  Delay 1000

color_change
ROBOT_STATE 1
algorithm_loop
  if ROBOT_STATE = 0
  do algorithm_execute
     Delay 50
  else if ROBOT_STATE = 1
  do algorithm_setup
     set ROBOT_STATE to 10
  else Delay 100

algorithm_interrupt
robot_interrupt_t interrupt , char * msg
```

8

Demonstration


Pera Swarm
Home
MQTT
Algorithm Generation
Dynamic Code Generation
Build
OTA Upload
Logic
Algorithm
Control Flow
Inputs
MQTT
Motors
Outputs
Variables
String
Math
Functions
ESP
algorithm_execute
algorithm_reset
algorithm_setup
algorithm_start
algorithm_stop
Serial Print
set ROBOT_STATE to
Delay 1000
NeoPixel colorwave
algorithm_interrupt

9

Challenges

Challenges and How overcame them

Challenges

- Dependency issue when building the existing firmware of robots.
- Google blockly visual programming tools doesn’t support c++ by default.
- Need to upload specific firmware to each robot by OTA.

Thanks!

Any questions?