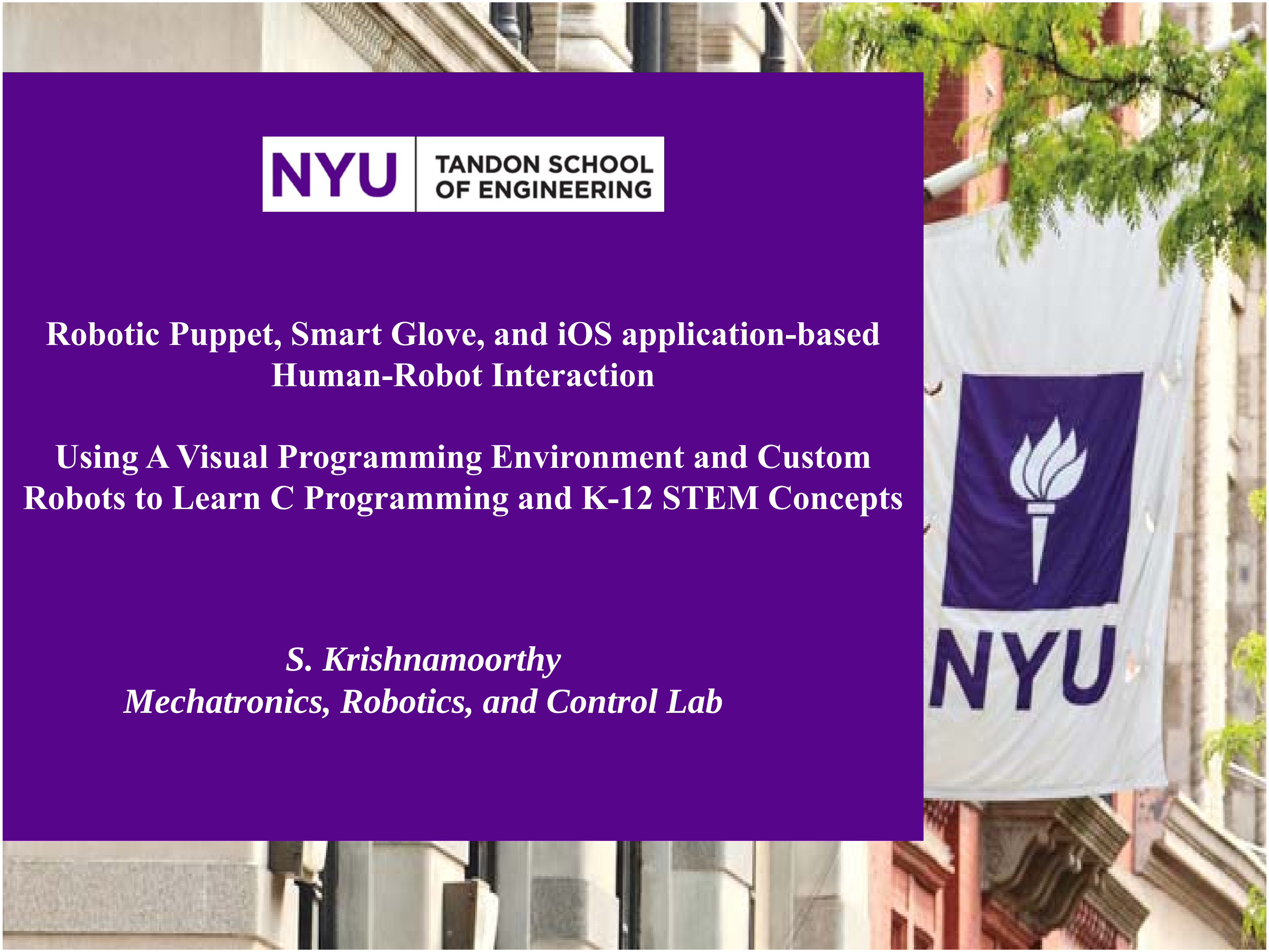


NYU TANDON SCHOOL OF ENGINEERING

Robotic Puppet, Smart Glove, and iOS application-based Human-Robot Interaction

Using A Visual Programming Environment and Custom Robots to Learn C Programming and K-12 STEM Concepts

S. Krishnamoorthy
Mechatronics, Robotics, and Control Lab

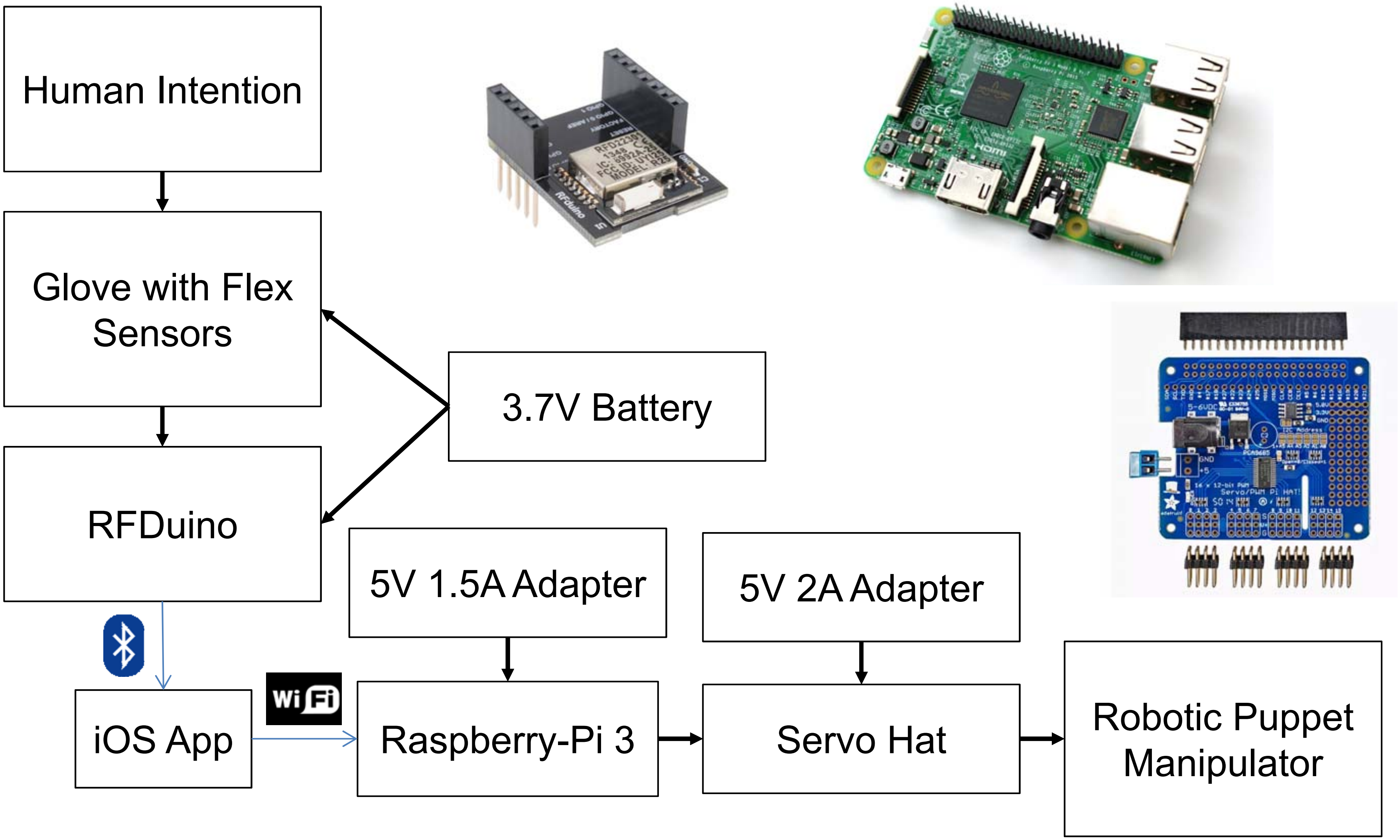
Motivation - 1

- Intuitive human-robot interaction
- Fluid and natural
- Leveraging technology
- Training Vs experience
- Use is performing arts

Robotic Puppet V2.0

- Rod puppet design
- 4 DOF “Sit-down” scenario
- Utilizes Bluetooth-LE and Wi-Fi
- Interfaced with ROS
- Glove interface configured via iOS application
- Websocket communication

Hardware Setup - 1



Human Intention

Glove with Flex Sensors

3.7V Battery

RFDuino

5V 1.5A Adapter

5V 2A Adapter

iOS App

Raspberry-Pi 3

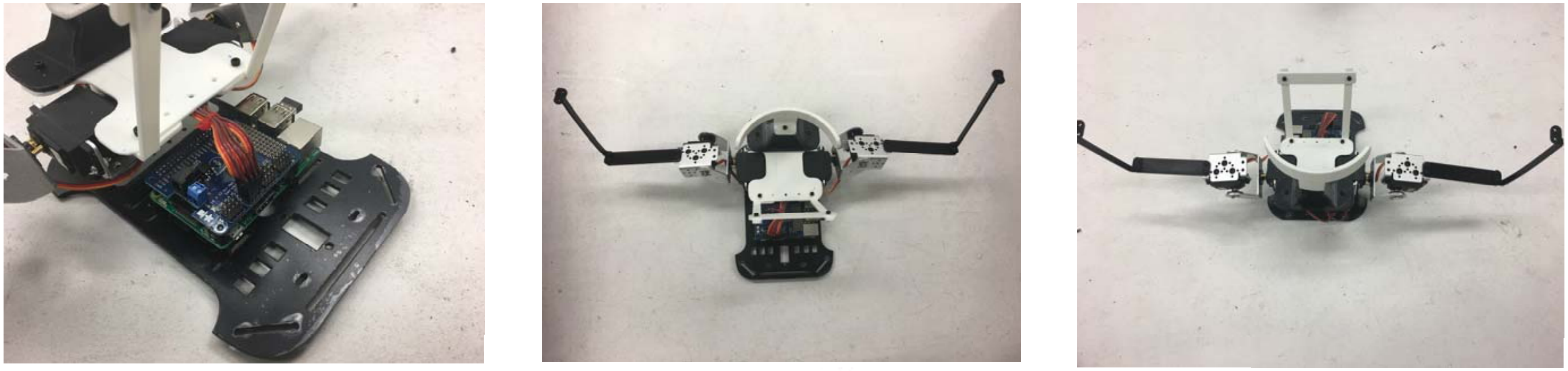
Servo Hat

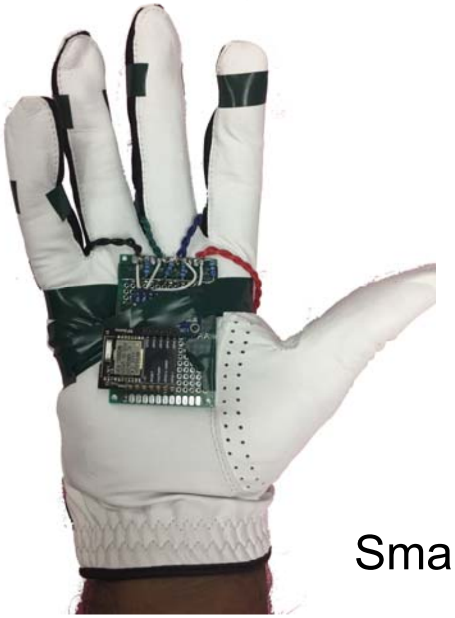
Robotic Puppet Manipulator

NYU

Hardware Setup - 2

Robotic Puppet Manipulator



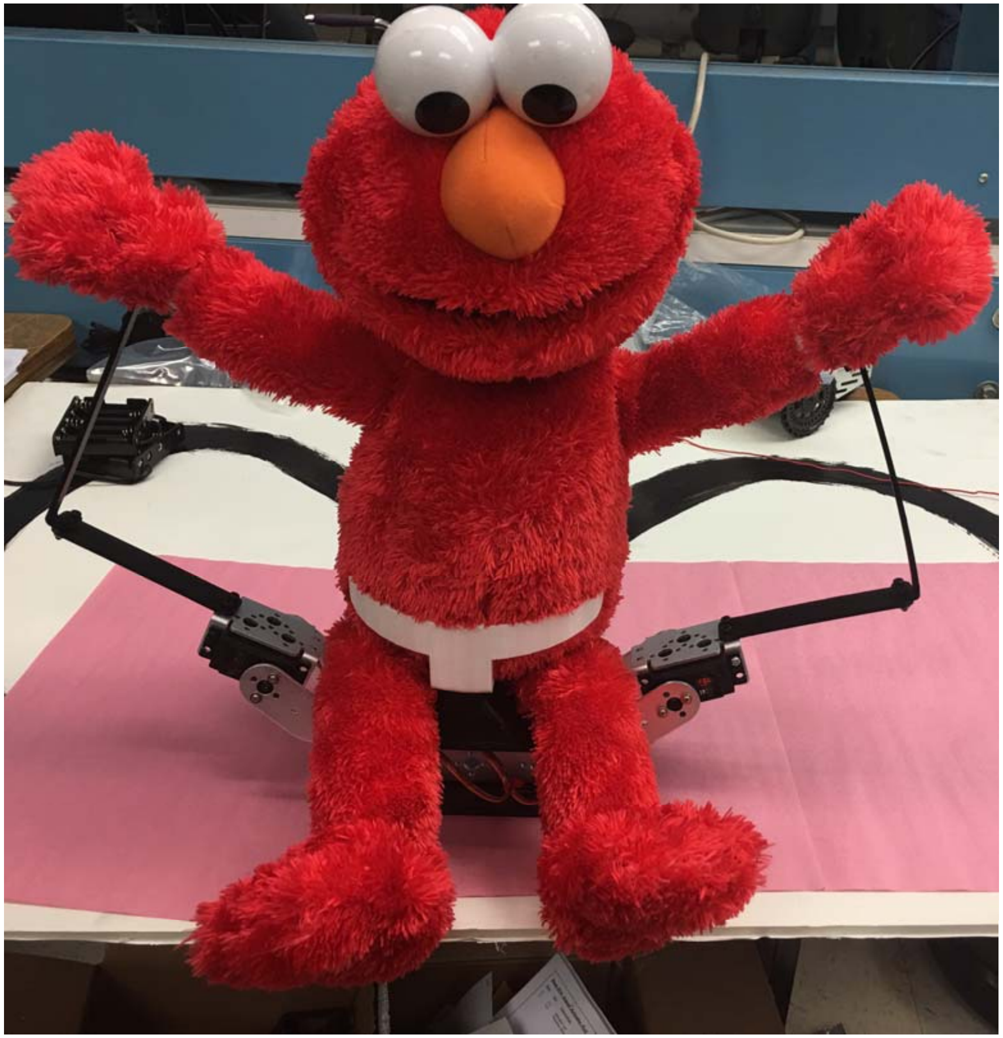


Smart Glove

Smart Glove

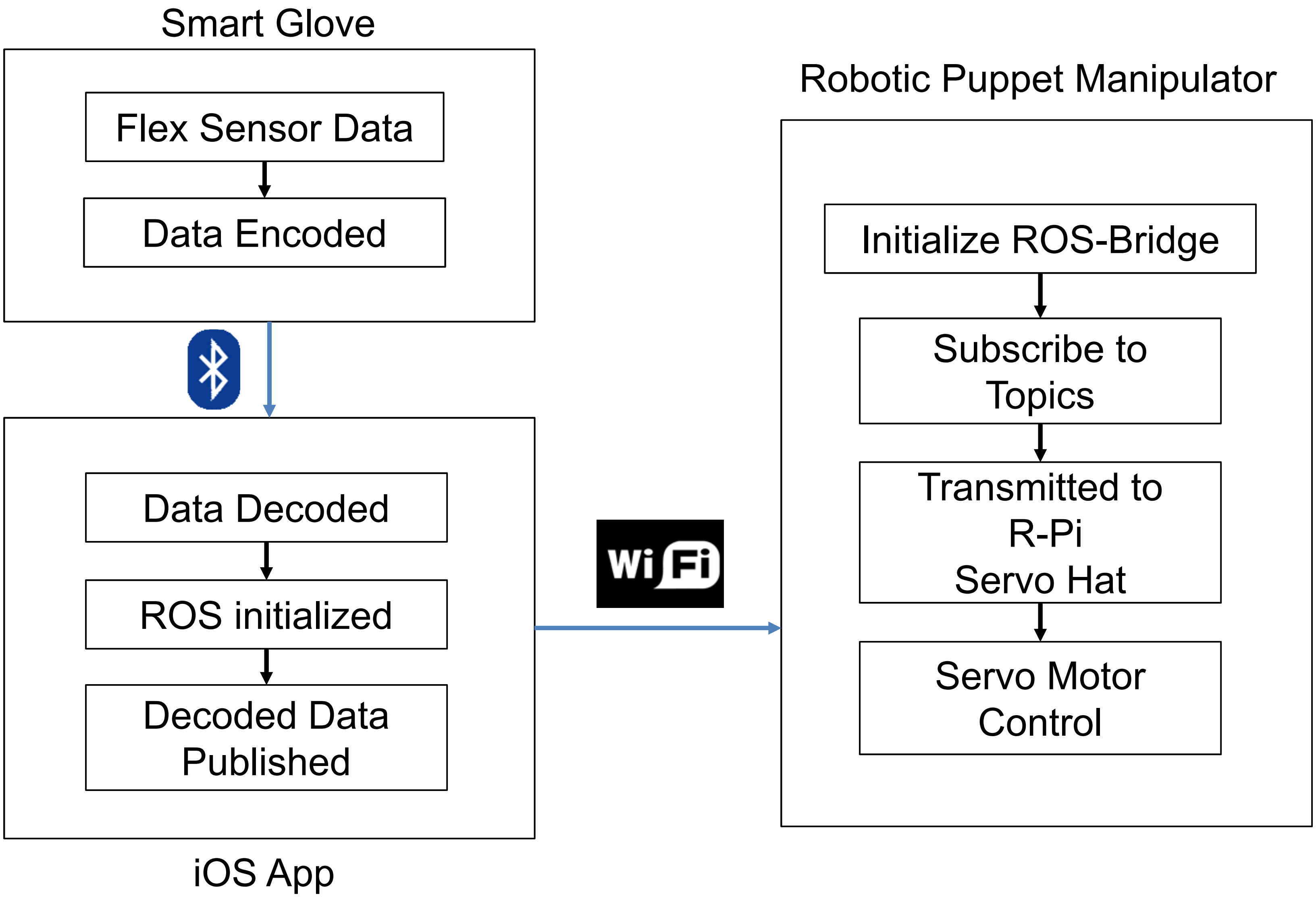
- Flex sensors measuring joint angle of MCP and PIP
- Proximal interphalangeal joints as primary
- 0-90° range
- Wireless connectivity via Rfduino BLE
- Powered by 3.7V Li-ion battery

Hardware Setup - 3



NYU

Software Setup - 1



Smart Glove

Flex Sensor Data

Data Encoded

Data Decoded

ROS initialized

Decoded Data Published

iOS App

Robotic Puppet Manipulator

Initialize ROS-Bridge

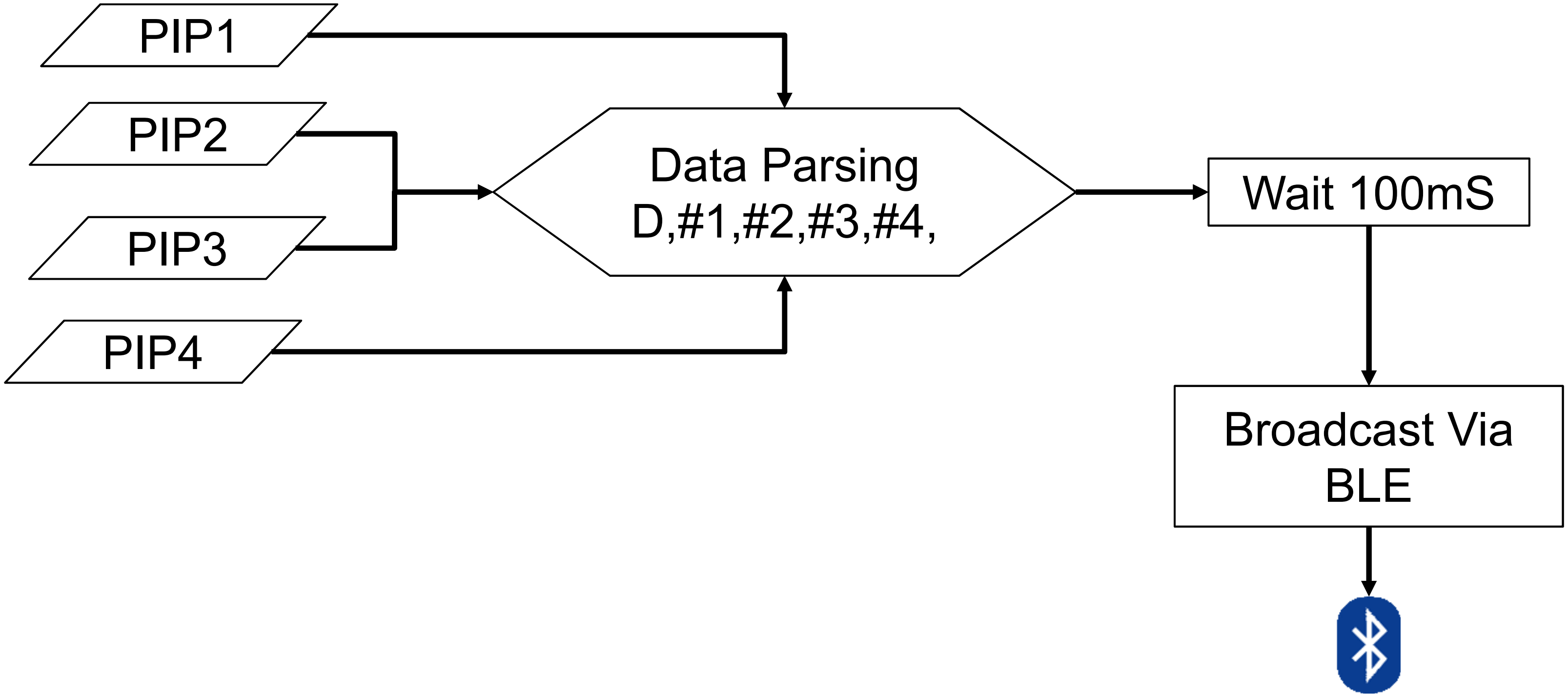
Subscribe to Topics

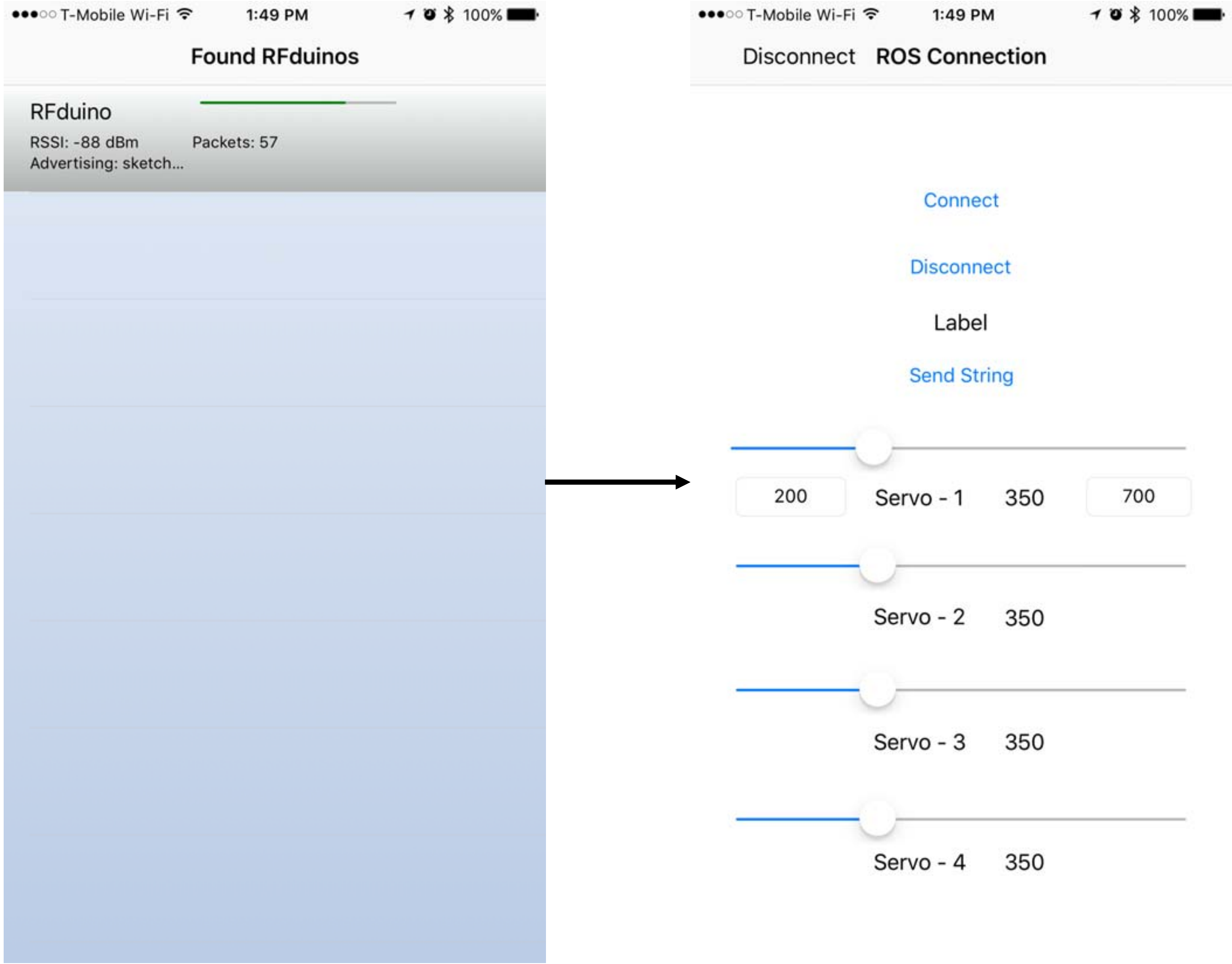
Transmitted to R-Pi Servo Hat

Servo Motor Control

Software Setup - 2

Smart Glove





●●●○○ T-Mobile Wi-Fi 1:49 PM 100%

Found RFduinos

RFduino

RSSI: -88 dBm Packets: 57

Advertising: sketch...

●●●○○ T-Mobile Wi-Fi 1:49 PM 100%

ROS Connection

200 Servo - 1 350 700

Servo - 2 350

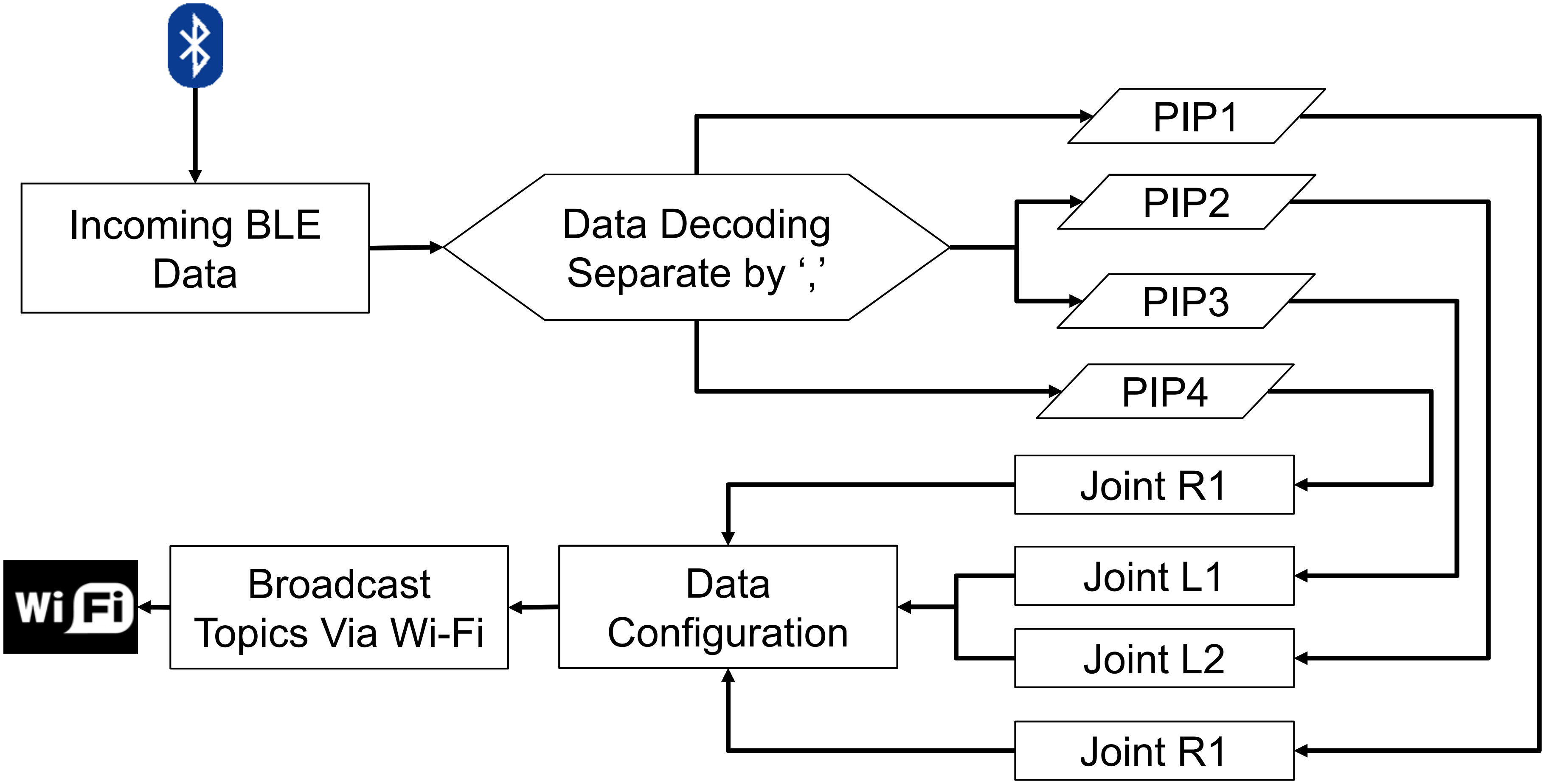
Servo - 3 350

Servo - 4 350

iOS Application

Software Setup - 3

iOS Application



Incoming BLE Data

Data Decoding Separate by ‘,’

PIP1

PIP2

PIP3

PIP4

Joint R1

Joint L1

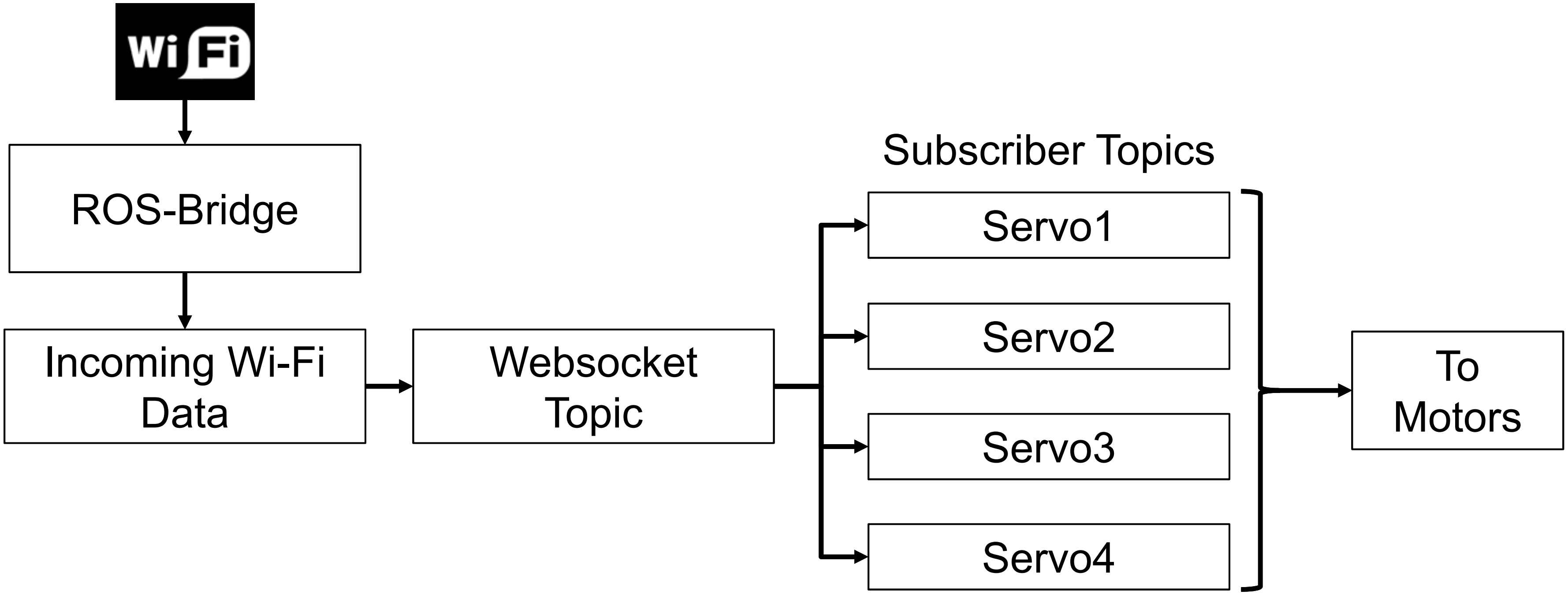
Joint L2

Joint R1

Data Configuration

Broadcast Topics Via Wi-Fi

Robotic Puppet Manipulator

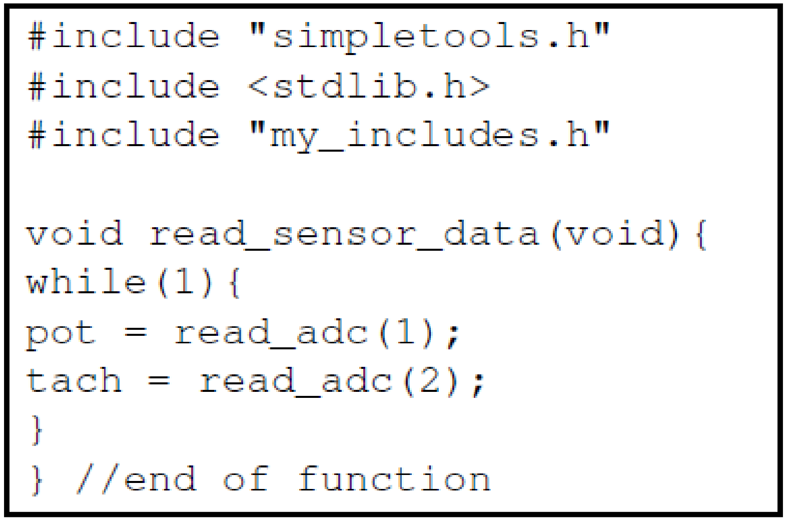


Motivation - 1

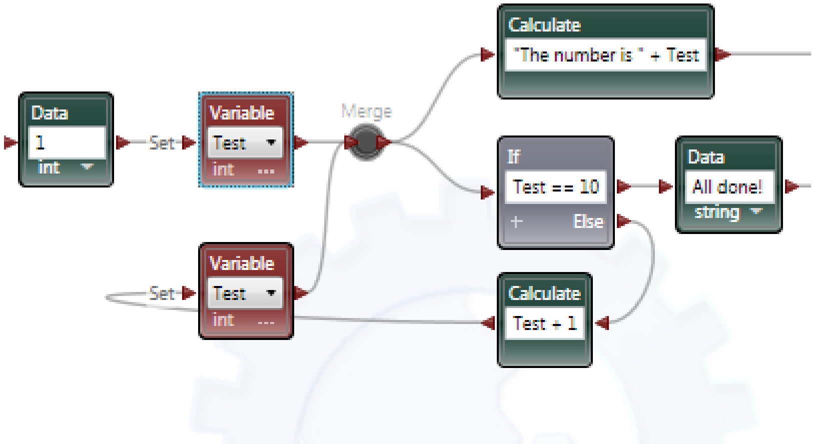
- Changes in K-12 education
- Integration of challenging concepts
- Increasing use of assistive learning tools
- Improve student engagement and participation
- Our Plan : To invent a novel learning tool with robots
- Cost-effective and easily deployable robotic platform
- Develop a lightweight software package

Visual v/s Text: Advantages

- Supports novice programmers
- No syntax errors
- Psychologically less daunting
- Offers a quick starting script
- Assists with unfamiliar APIs



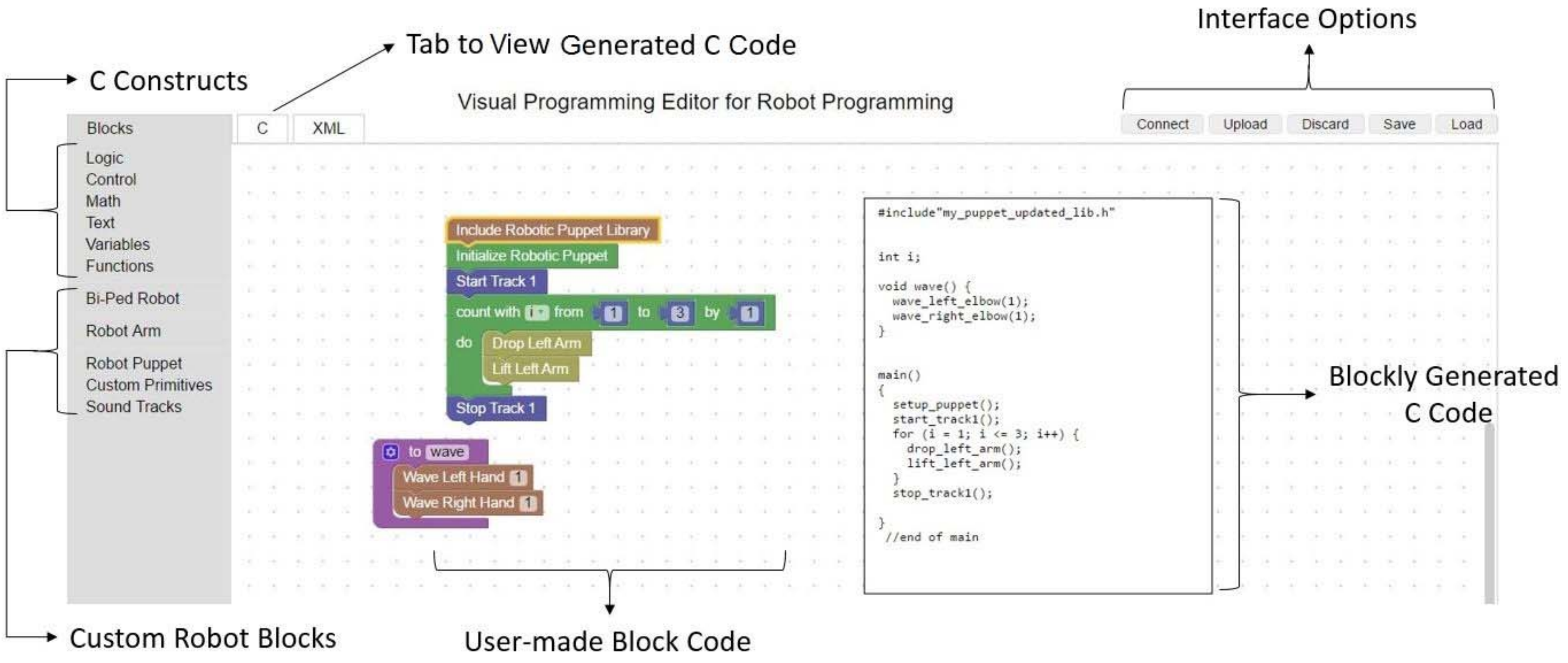
```
#include "simpletools.h"
#include <stdlib.h>
#include "my_includes.h"

void read_sensor_data(void){
while(1){
pot = read_adc(1);
tach = read_adc(2);
}
} //end of function
```



Software Package - Using Blockly

- Visual editor for programming
- Developed by Google
- Access to basic and advanced blocks
- Easy to customize and integrate into new applications
- Produces Java/Python code
- Modified to generate C code

Blockly User Interface

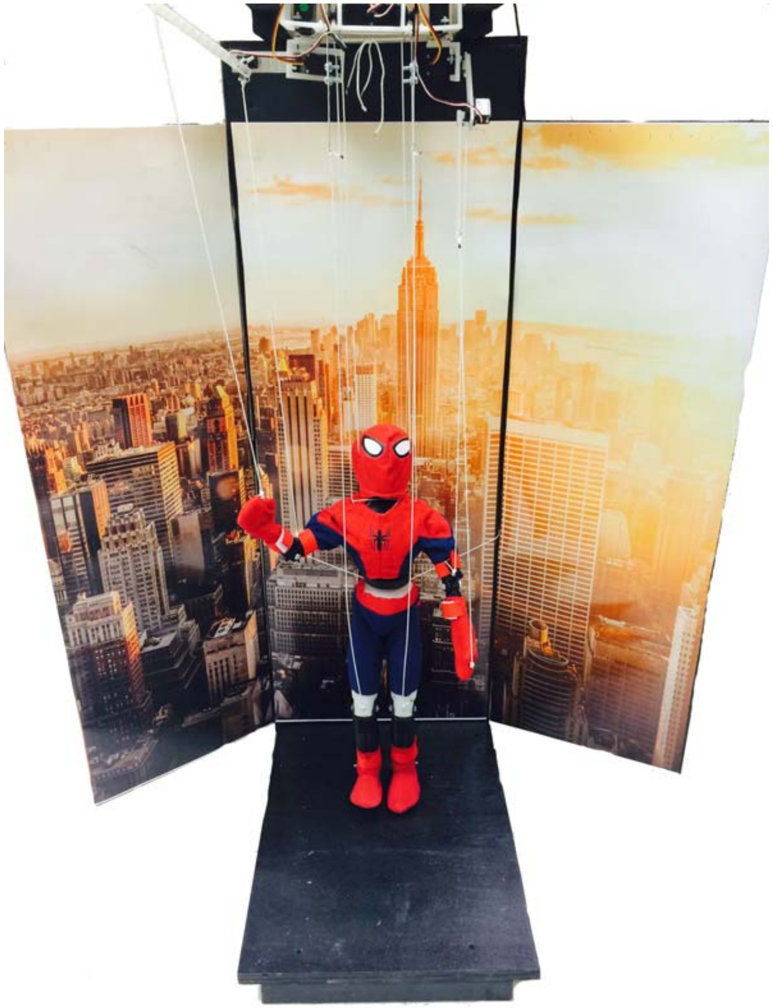


```
#include"my_puppet_updated_lib.h"


int i;

void wave() {
  wave_left_elbow(1);
  wave_right_elbow(1);
}


main()
{
  setup_puppet();
  start_track1();
  for (i = 1; i <= 3; i++) {
    drop_left_arm();
    lift_left_arm();
  }
  stop_track1();

}
 //end of main
```

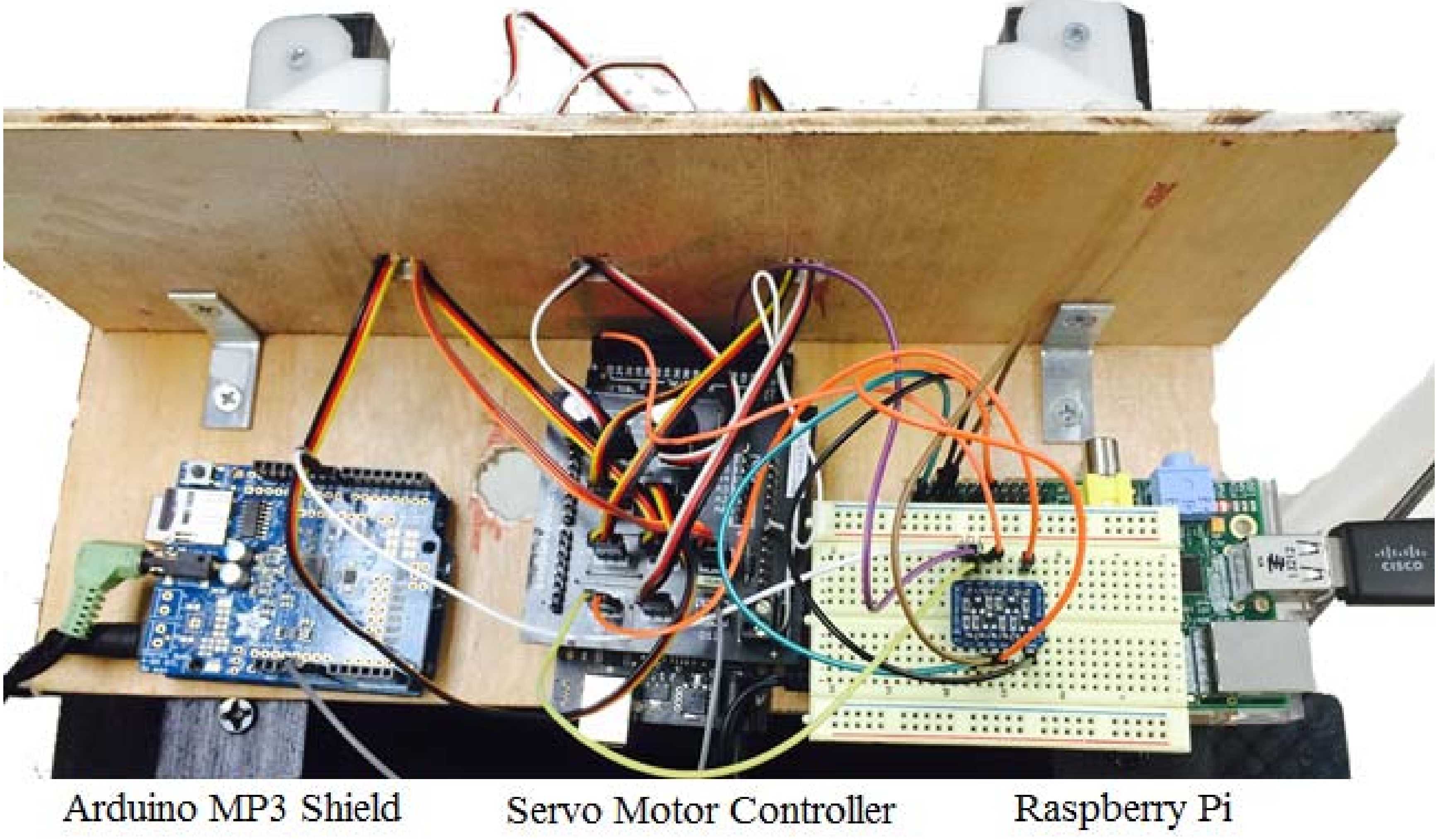
NYU

Robotic Puppet

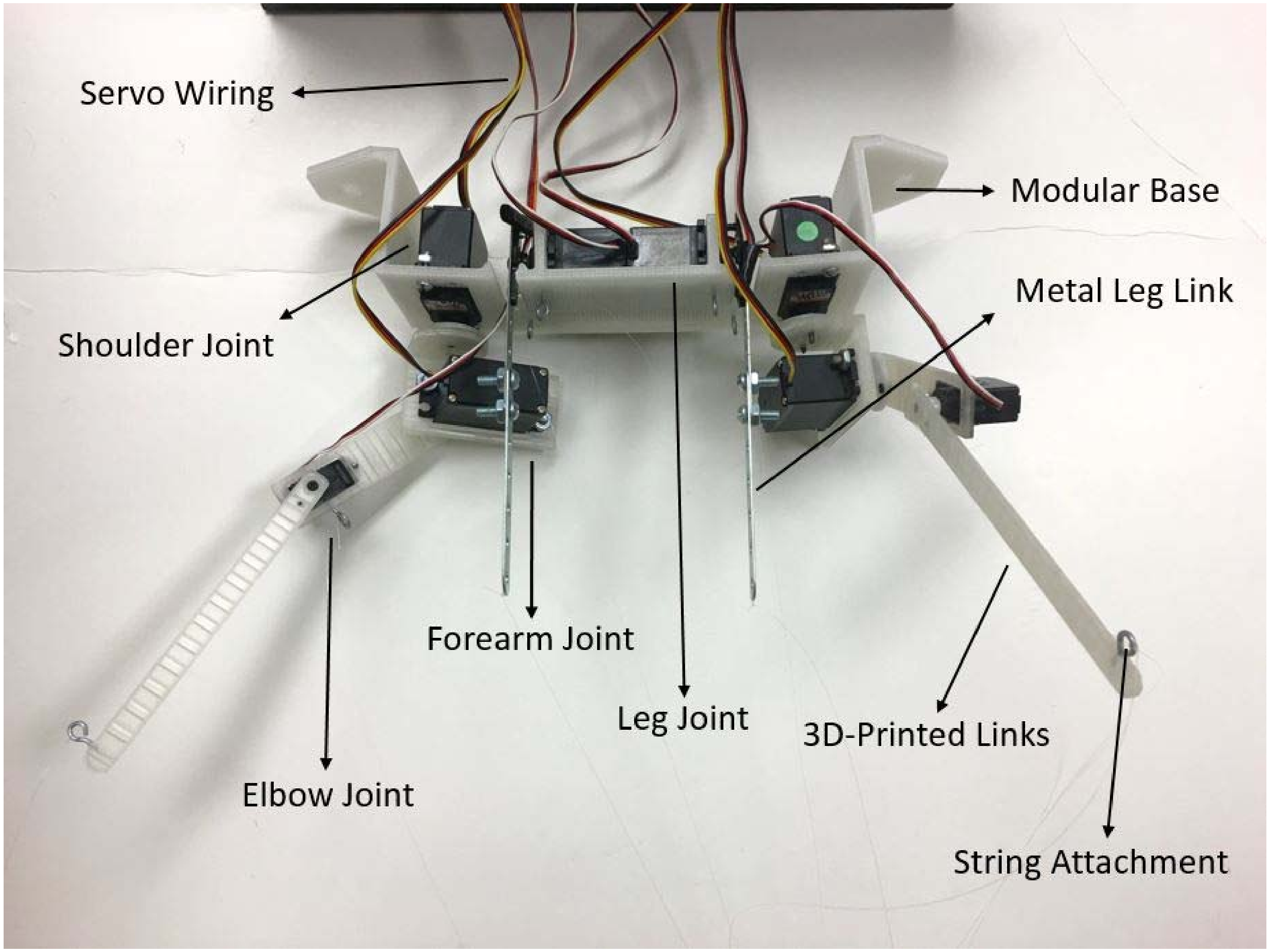
- Uncommon implementation
- 3D-printed DIY platform
- Fusion of art with engineering
  - Dance, music, etc.
- Simplified approach to create stories
  - Distinct primitives are represented as blocks
  - Ability to increase block library size
  - Multiple combinations for storylines
  - Control multiple puppets



Embedded Hardware for Puppet

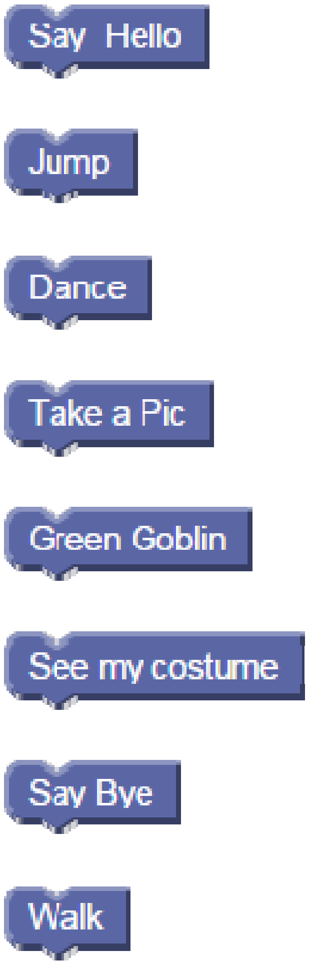


Puppet Manipulator Setup



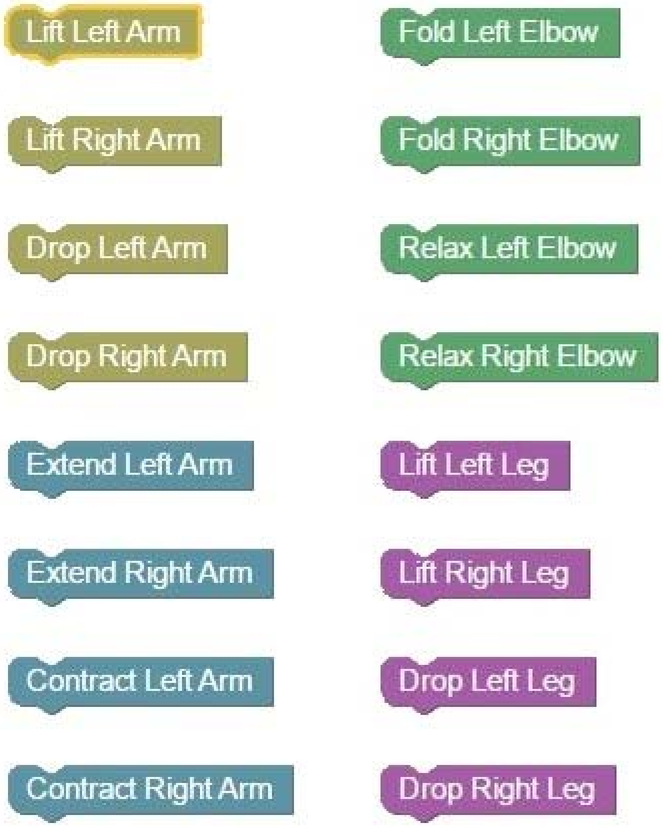
Robotic Puppet Motion Primitives

Choreographed Sequences



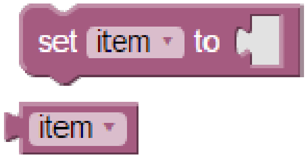
- Say Hello
- Jump
- Dance
- Take a Pic
- Green Goblin
- See my costume
- Say Bye
- Walk

Joint Primitives



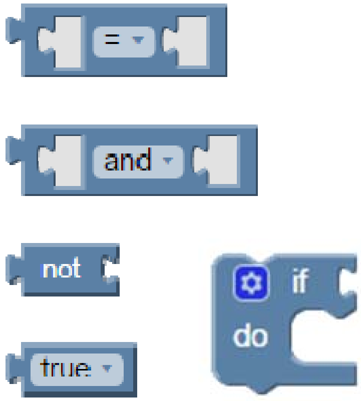
- Lift Left Arm
- Lift Right Arm
- Drop Left Arm
- Drop Right Arm
- Extend Left Arm
- Extend Right Arm
- Contract Left Arm
- Contract Right Arm
- Fold Left Elbow
- Fold Right Elbow
- Relax Left Elbow
- Relax Right Elbow
- Lift Left Leg
- Lift Right Leg
- Drop Left Leg
- Drop Right Leg

Variables



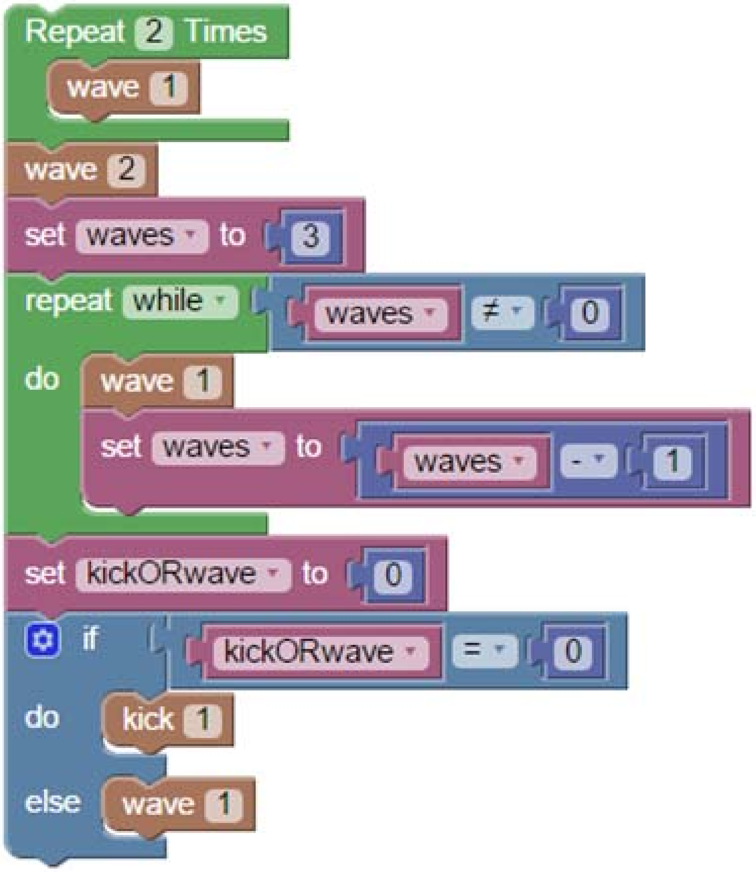
- set item to
- item

Logical operations

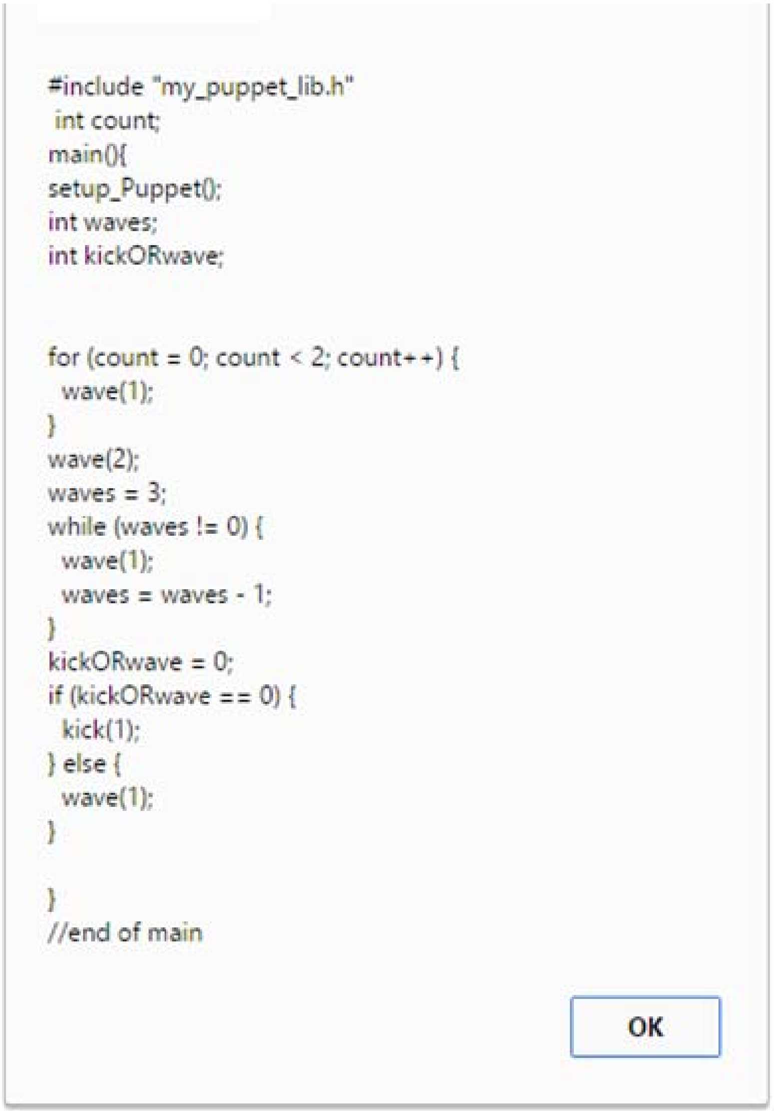


- =
- and
- not
- true
- if do

Robotic Puppet: Example Code



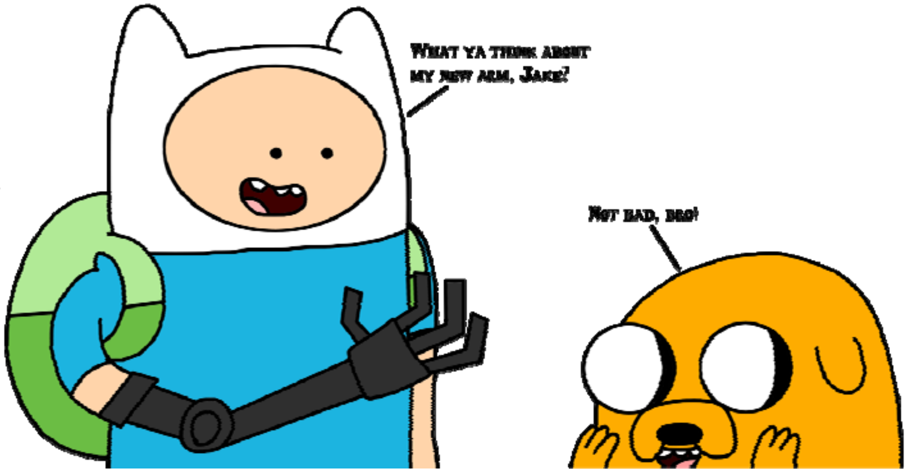
```
Repeat 2 Times
    wave 1
wave 2
set waves to 3
repeat while waves ≠ 0
do  wave 1
    set waves to waves - 1
set kickORwave to 0
if kickORwave = 0
do  kick 1
else wave 1
```



```
#include "my_puppet_lib.h"
 int count;
main(){
setup_Puppet();
int waves;
int kickORwave;


for (count = 0; count < 2; count++) {
  wave(1);
}
wave(2);
waves = 3;
while (waves != 0) {
  wave(1);
  waves = waves - 1;
}
kickORwave = 0;
if (kickORwave == 0) {
  kick(1);
} else {
  wave(1);
}

}
//end of main
```

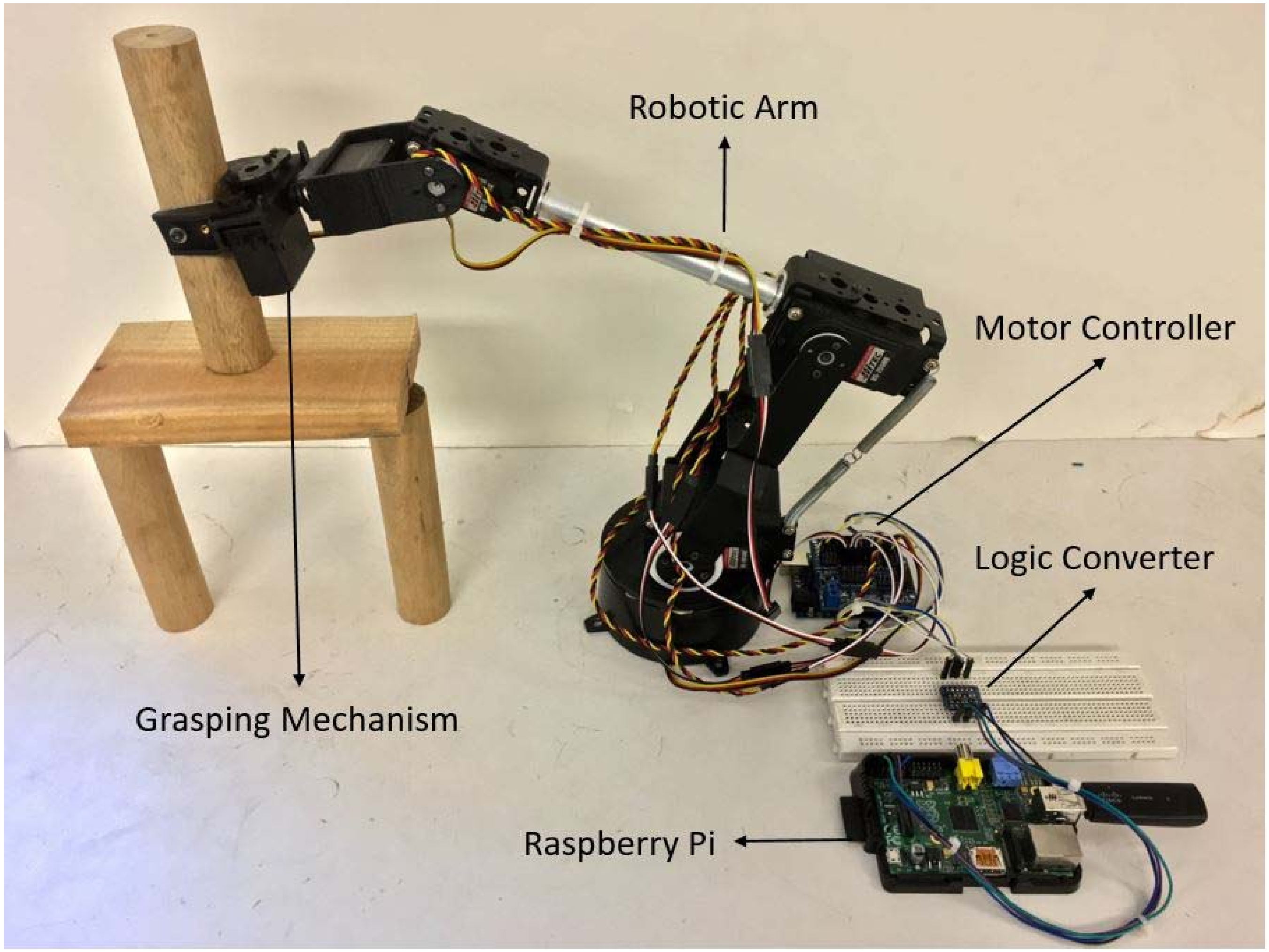
Robotic Arm

- Presents real-world applications
- Reliable platform for introduction to robotics
- Widely available and easy to implement
- Cost-effective
- Able to explore wide variety of STEM concepts
- Invokes interest in young students



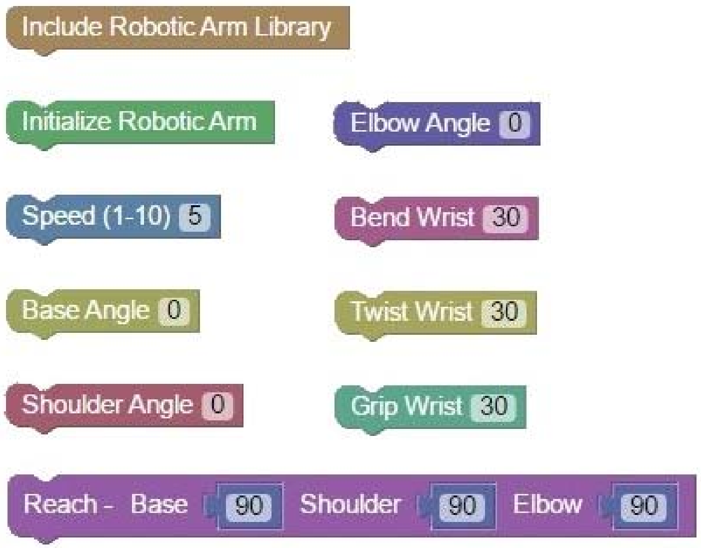
NYU

Robotic Arm



Robotic Arm Primitives

Joint Primitives



Include Robotic Arm Library

Initialize Robotic Arm

Speed (1-10) 5

Base Angle 0

Shoulder Angle 0

Elbow Angle 0

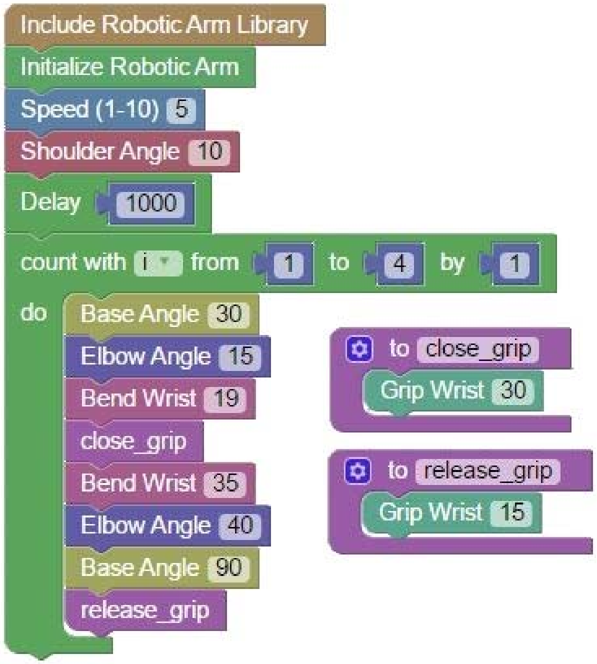
Bend Wrist 30

Twist Wrist 30

Grip Wrist 30

Reach - Base 90 Shoulder 90 Elbow 90

Example Program



Include Robotic Arm Library

Initialize Robotic Arm

Speed (1-10) 5

Shoulder Angle 10

Delay 1000

count with i from 1 to 4 by 1

do

Base Angle 30

Elbow Angle 15

Bend Wrist 19

close_grip

Bend Wrist 35

Elbow Angle 40

Base Angle 90

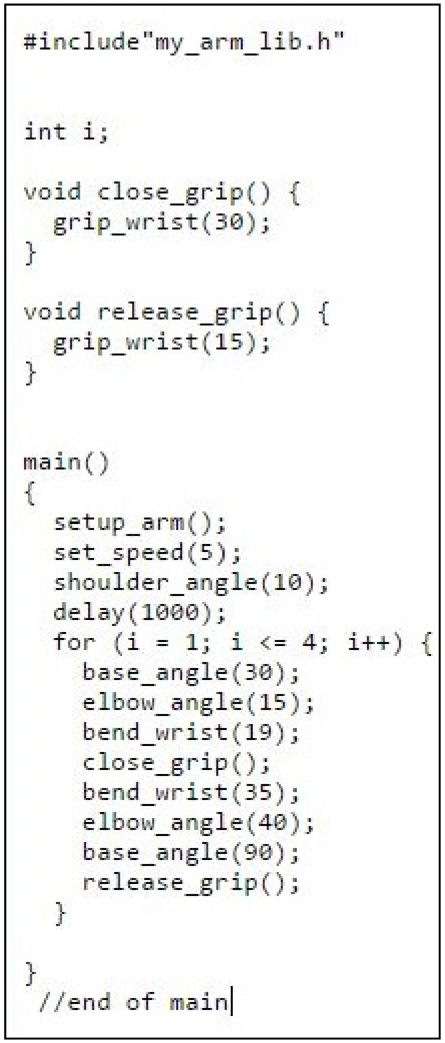
release_grip

to close_grip

Grip Wrist 30

to release_grip

Grip Wrist 15

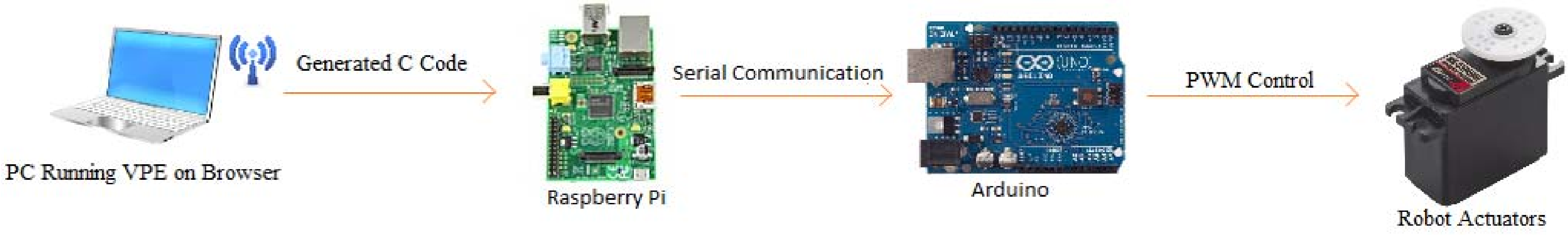


```
#include"my_arm_lib.h"


int i;

void close_grip() {
  grip_wrist(30);
}

void release_grip() {
  grip_wrist(15);
}


main()
{
  setup_arm();
  set_speed(5);
  shoulder_angle(10);
  delay(1000);
  for (i = 1; i <= 4; i++) {
    base_angle(30);
    elbow_angle(15);
    bend_wrist(19);
    close_grip();
    bend_wrist(35);
    elbow_angle(40);
    base_angle(90);
    release_grip();
  }

}
 //end of main
```

NYU

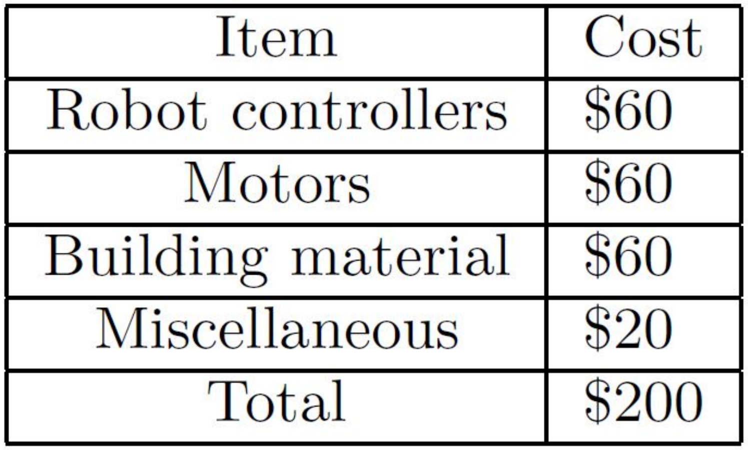
System Architecture

1. User creates block program on VPE
2. Blockly generates corresponding C code
3. Code transferred to R-Pi and is compiled
4. Movement commands sent to Arduino board
5. Arduino relays signals to individual motors



Cost of Implementation

- 3D-printed robot parts
- Off-the-shelf electronic components
- Free to use software package
- Hardware can be upgraded/downgraded easily
- Components are reusable



Item	Cost
Robot controllers	$60
Motors	$60
Building material	$60
Miscellaneous	$20
Total	$200

Lesson Selection and Design

- Designed in collaboration with teachers
- K-5, 6-8, and 9-12 grade levels
- Includes STEM, computer programming and problem solving tasks
- Molded to explore concepts through short activities
- Inclusion of real-world applications
- Practical and experimental learning experience

NYU

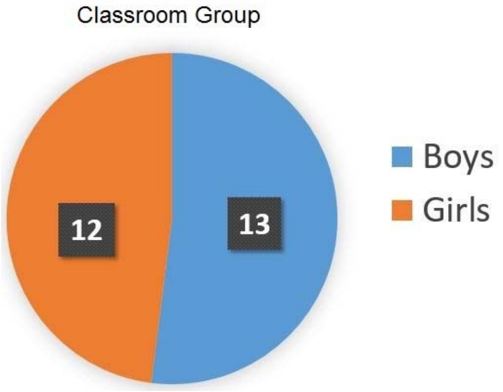
Activities

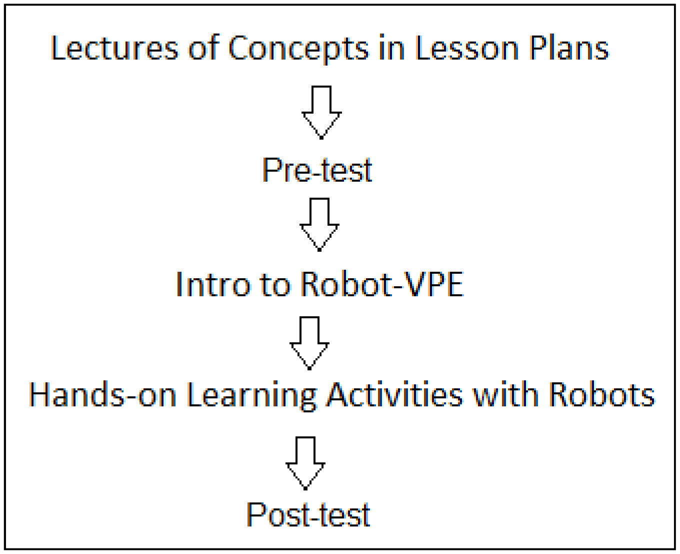
Robots → Grade Levels ↓	Robotic Puppet	Robotic Arm
K-5	• Human VS robot • Muscle VS motor • Basic math • Designing a hand-shake robot	• Movement and trajectory • Links and joints • Distance and measurement
6-8	• Human VS robot anatomy • Range of motion and constraints • Programming sequential movements with playback voice	• Angles and measurements • Gravity and center of mass • “Demolition” activity
9-12	• Robot design • Advanced programming constructs • Synchronized choreography • Actuation mechanism design	• Robot mechanics and characteristics • Measurement and trajectory planning • “Center of mass” activity • “Bridge construction” activity

NYU

Participation and Assessment

- Classroom integration for practical learning
- Tested by students from 5-10 grades
- Volunteered study and testing





Lectures of Concepts in Lesson Plans

Pre-test

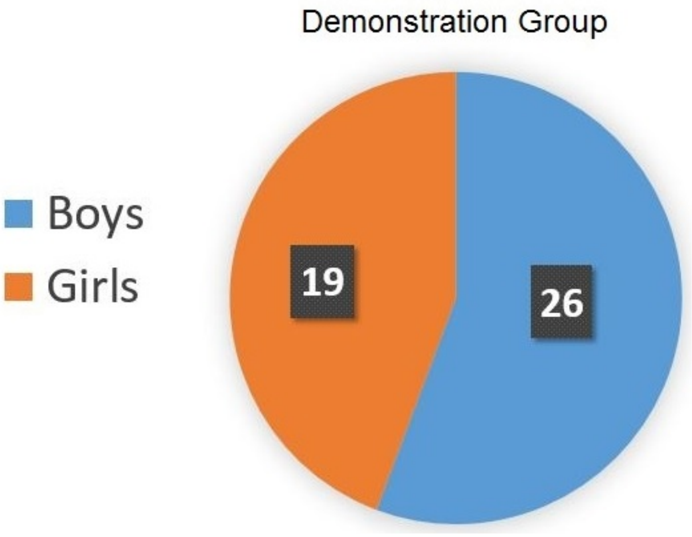
Intro to Robot-VPE

Hands-on Learning Activities with Robots

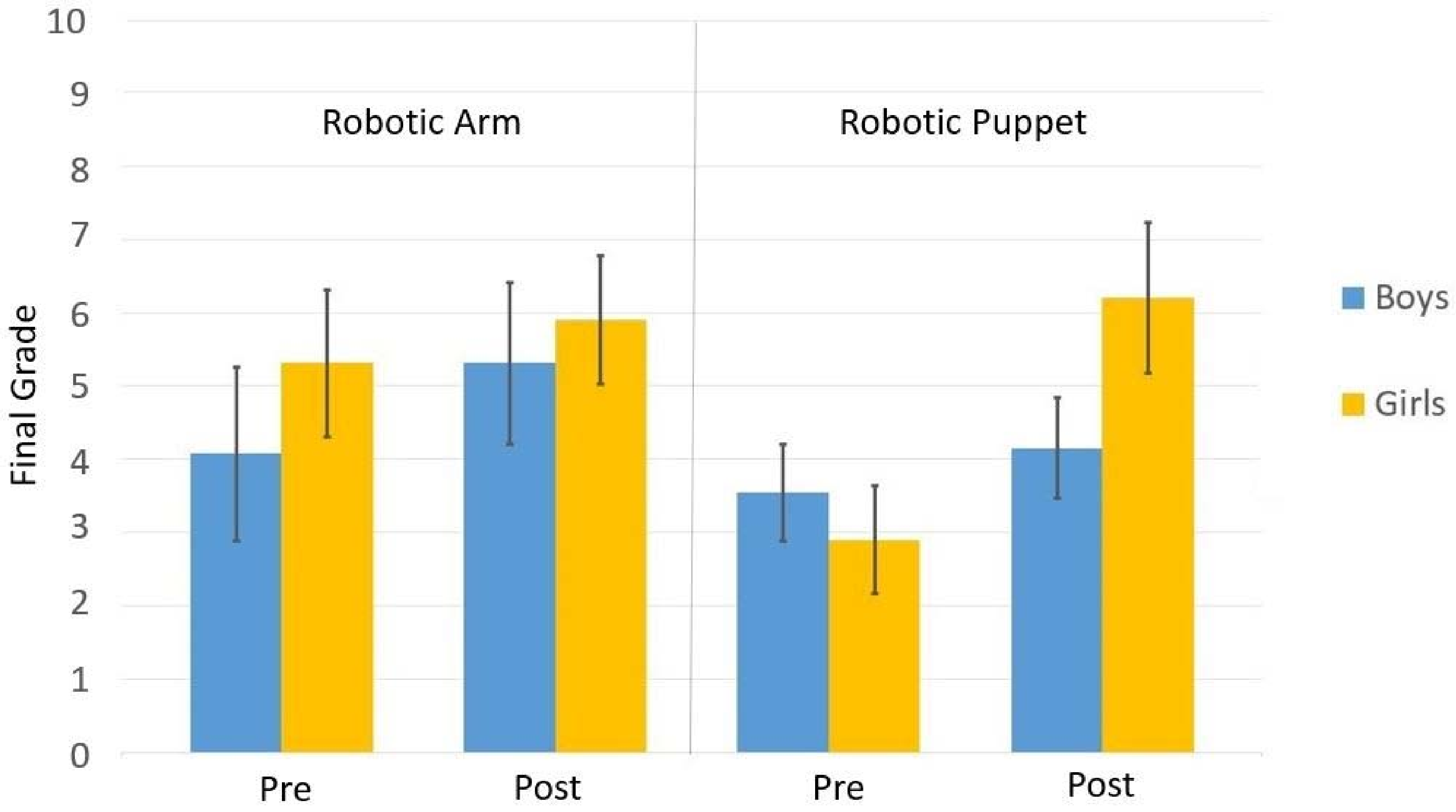
Post-test

Demonstration

- Tested by students from various grade levels, teachers, parents and educators
- Included evaluation of at least one activity with robot-VPE system



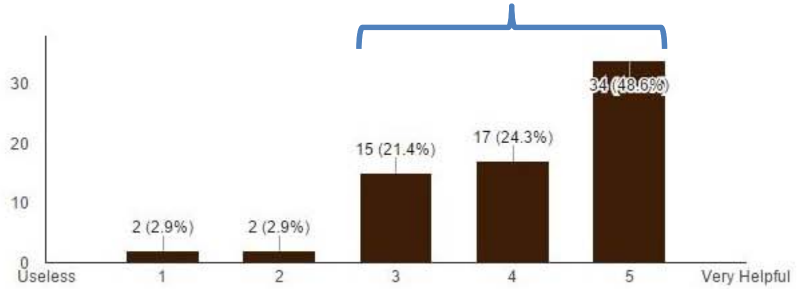
Observations and Results - 1



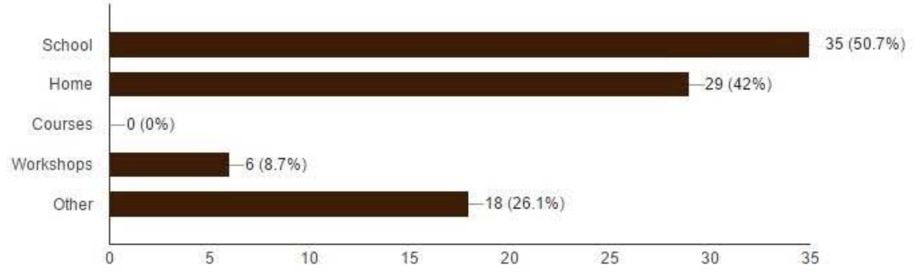
Observations and Results - 2

How helpful was Blockly in understanding the flow of the program? (70 responses)

>95%

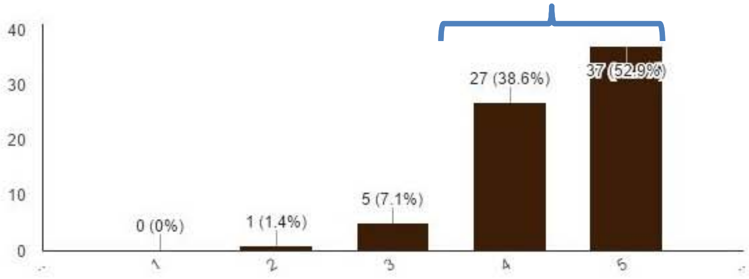


Where do you learn programming? (69 responses)

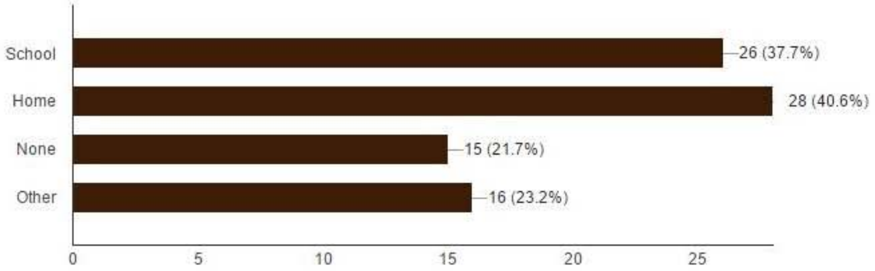


How engaging was the programming activity? (70 responses)

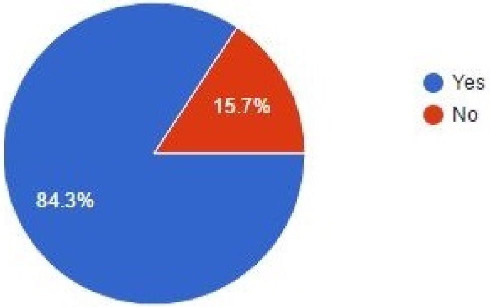
>90%



Did you play with robots before? (69 responses)



Will you accept Blockly tool as your programming method?



Conclusion and Future Work

- Robot-VPE interface as a platform for interactive experiments and practical learning of K-12 concepts
- Hands-on robotics and programming experiences
- Improved understanding of subject
- Cost effective and easily configurable DIY solution
- Scope for deployment and integration into classrooms
- Web-camera based setup for remote experiments

NYU

Acknowledgements

DRK-12 DRL: 1417769
GK-12 DGE: 0741714
RET Site EEC-1132482

Brooklyn Community Foundation (BCF)

Xerox Corporation

Motorola Innovation Generation Grant

J.P. Morgan Chase Foundation

Black Male Donor Collaborative

NY Space Grant Consortium

White Cedar Fund

Appendix

Sample questions:

K-5:

1. Design program to make the robot kick 2+2 times and wave 5-2 times
2. Count the number of joints in your arm and compare with the robotic arm.

6-8:

1. Inspect the range of motion of your arm and determine your workspace. Repeat the same with the Robotic Arm.
2. Choreograph a scene from your favorite movie for the puppet and provide playback voice. You may use all available programming blocks.

9-12:

1. Determine the trajectory required by the robot to pick the block while avoiding obstacles. Program the path.
2. Program the Robotic Puppet to dance to your favorite song while using *for, while, if*, and *functions*.