ASEE Zone 1
Spring Conference

How Raspberry Pi Technology Can Enhance Students' Learning Opportunities in Technology

by Jim Mallory and Ed Lucas
RIT/NTID

Agenda

- Definition - What is Raspberry pi?
- History
- Our Project
  - Phase 1 - RPi + Brick Pi Interface
  - Phase 2 - RPi + Pi Storms Interface
- Skills Developed
- Q & A

Raspberry Pi - Definition

- An inexpensive, ubiquitous, Linux based platform that will change how we do business in the educational technology field.
- Credit-card sized computer - costs $35.
- Plugs into a computer monitor or TV.
- Uses a standard keyboard and mouse.

Raspberry Pi - History

- Invented by a tiny UK charity
- First Released 2/2012
- 19,000,000+ sold (as of 3/2018)
- Original Target Market
- Outside Markets Unanticipated
- Manufactured in Pencoed, Wales
  - (some China and Japan)
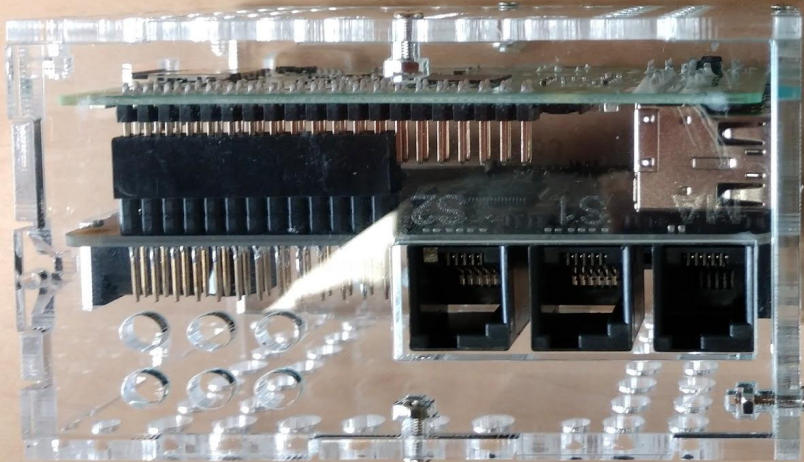 - Current RPi = Raspberry Pi 3 Model B+

Our Project Goal

ACT Capstone Students - Design and develop User Friendly Activity for young or non-techie students to learn programming

# Modules

Project Requirements

- Hardware
  - Raspberry Pi
    - Brick Pi - Lego Interface
    - Pi Storms - Lego Interface
    - Peripheral Devices (Motors, etc.)
- Software
  - Raspian Pi - OS Environment
  - Python - high level coding
  - Scratch, Blockly - easy to learn, teach students how to program

Project Phase 1 Brick Pi

- Replace Lego NXT (Left) with equivalent RPi Design (Right)

Design Mistakes = Learning Opportunities
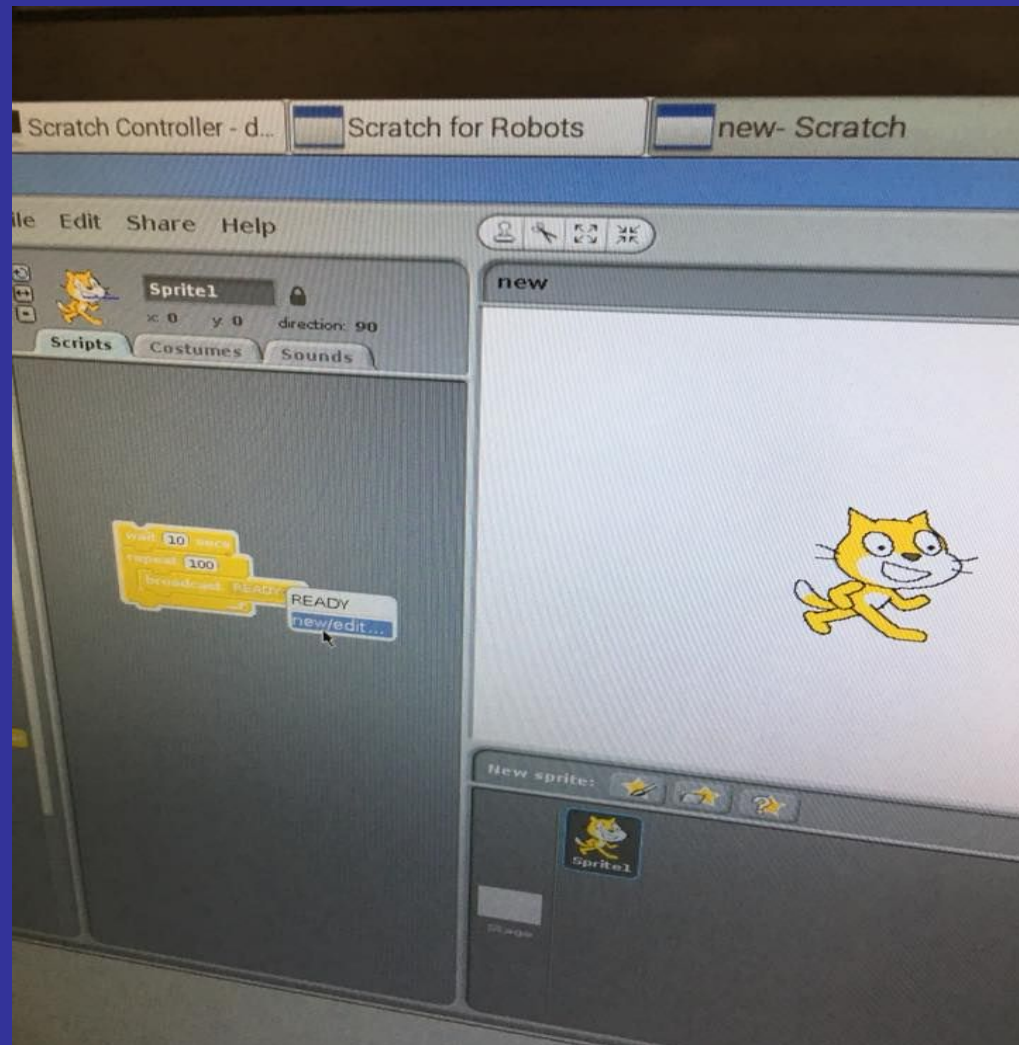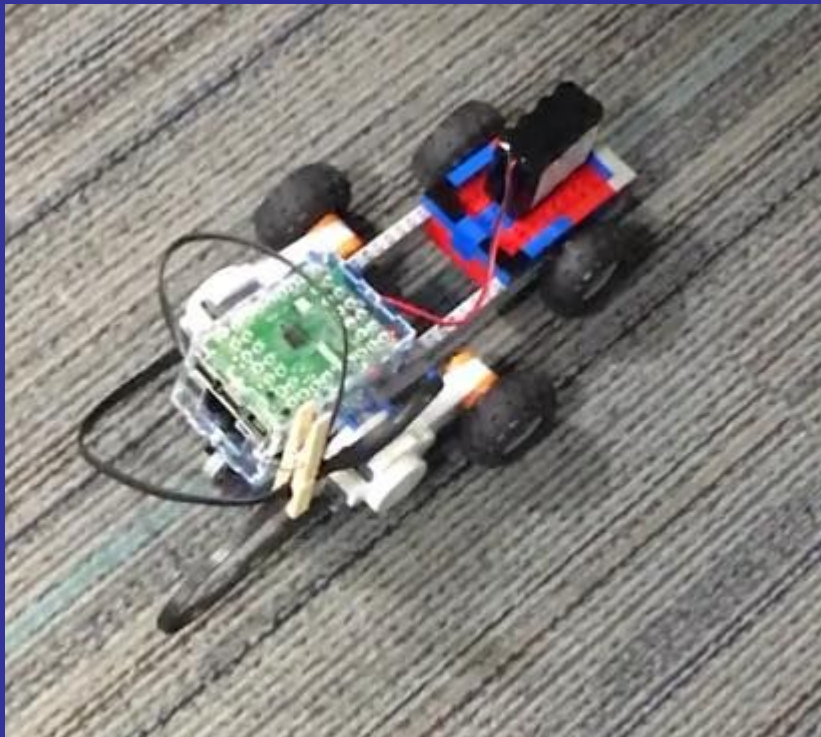
Design Mistakes = Learning Opportunities

Out-of-the-Box Thinking, New Skill Development

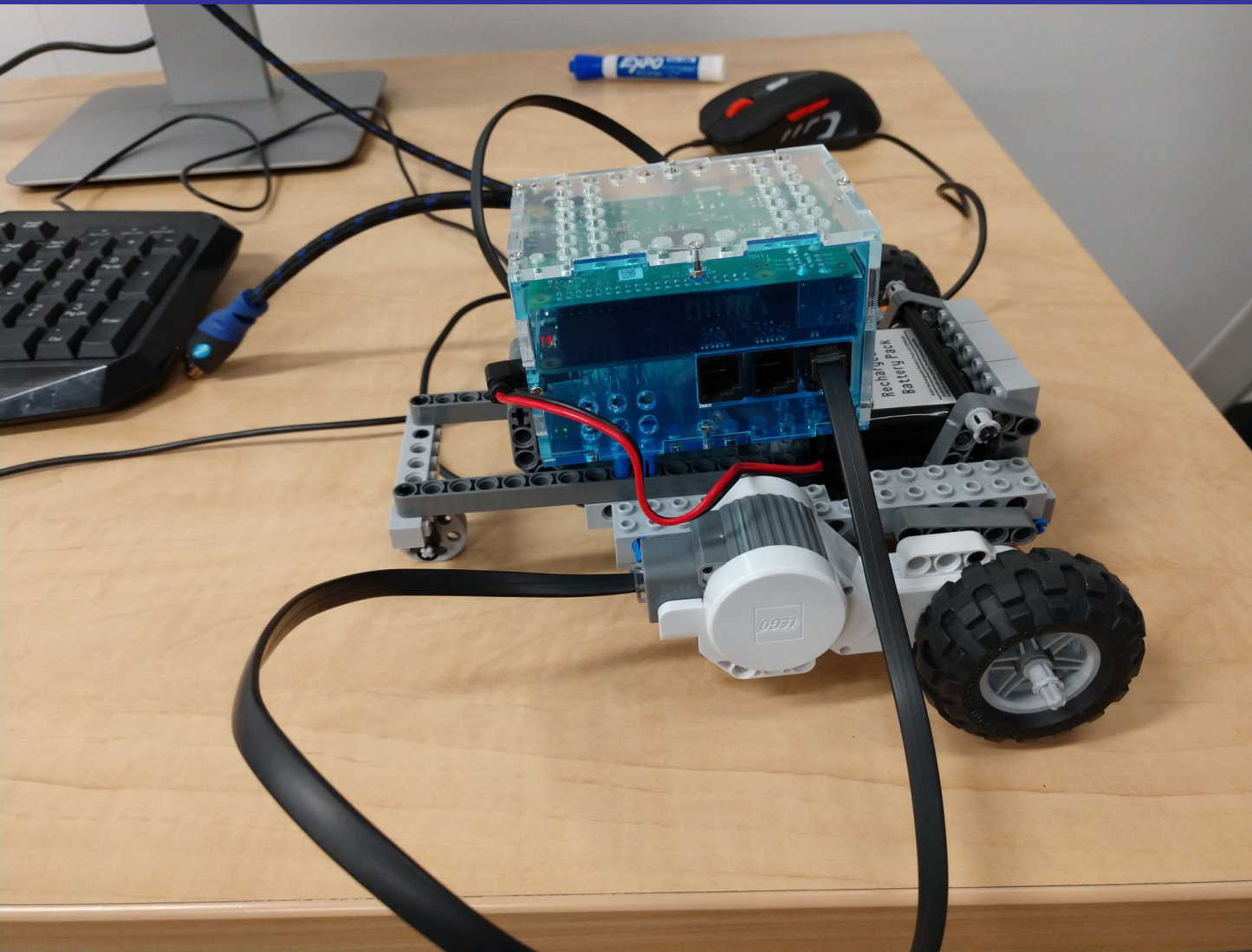# Programming Using Scratch

Testing on Non-Techie Students

Testing on Non-Techie Students

Phase 1 - Final Working Prototype

Phase 2 - Pi Storms

- Unlike Brick Pi, PiStorm has built-in WiFi
- Has touch screen controls
- Programming and control via web interface.
- Monitor and keyboard or USB not required
- Program in Blockly graphical environment or in Python

Phase 2 - Pi Storms

- Access PiStorms unit via web interface or secure shell
- Highly Customizable
- Portable

PiStorms Control Interface

Essentially a web server running on the Pi

Python Programming

```python
 9   # This program is distributed in the hope that it will be useful,
10   # but WITHOUT ANY WARRANTY; without even the implied warranty of
11   # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.    See the
12   # GNU General Public License for more details.
13   #
14   # You should have received a copy of the GNU General Public License
15   # along with this program; if not, write to the Free Software
16   # Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
17   #
18   #mindsensors.com invests time and resources providing this open source code,
19   #please support mindsensors.com  by purchasing products from mindsensors.com!
20   #Learn more product option visit us @  http://www.mindsensors.com/
21
22   import os, socket
23   import ConfigParser
24   from PiStorms import PiStorms
25   psm = PiStorms()
26
27   config = ConfigParser.RawConfigParser()
28   config.read("/usr/local/mindsensors/conf/msdev.cfg")
29   homefolder = config.get("msdev", "homefolder")
30 * try:
31 *     with open(os.path.join(homefolder, ".version"), "r") as f:
32           version_no = f.readline().strip()
33 * except IOError:
34       version_no = "unknown"
35
36   psm.screen.drawDisplay("About Me")
37   psm.screen.termPrintln("Device: {}".format(psm.GetDeviceId().rstrip("\0")))
38   psm.screen.termPrintln("Feature: {}".format(psm.psc.GetDeviceFeatures().rstrip("\0")))
39   psm.screen.termPrintln("f/w version: {}".format(psm.GetFirmwareVersion().rstrip("\0")))
40   psm.screen.termPrintln("s/w version: {}".format(version_no))
41   psm.screen.termPrintln("Hostname: {}".format(socket.gethostname()))
42   psm.screen.termPrintln("Battery: {}V".format(psm.battVoltage()))
43
44 * def getIP(iface):
45       ip = os.popen('ifconfig {} | grep "inet addr" | cut -d: -f2 | cut -d" " -f1'.format(iface)).read().rstrip()
46       return ip if ip != '' else "not present"
47 * def updateNetworkInfo():
48       psm.screen.termPrintAt(5, "eth0: {}".format(getIP("eth0")))
49       psm.screen.termPrintAt(6, "wlan0: {}".format(getIP("wlan0")))
50   psm.untilKeyPressOrTouch(updateNetworkInfo)
51
52   psm.screen.termPrintAt(8, "Exiting to menu")
53
```

Blockly Programming

Implementing User Friendly Activity at SVP Sampling
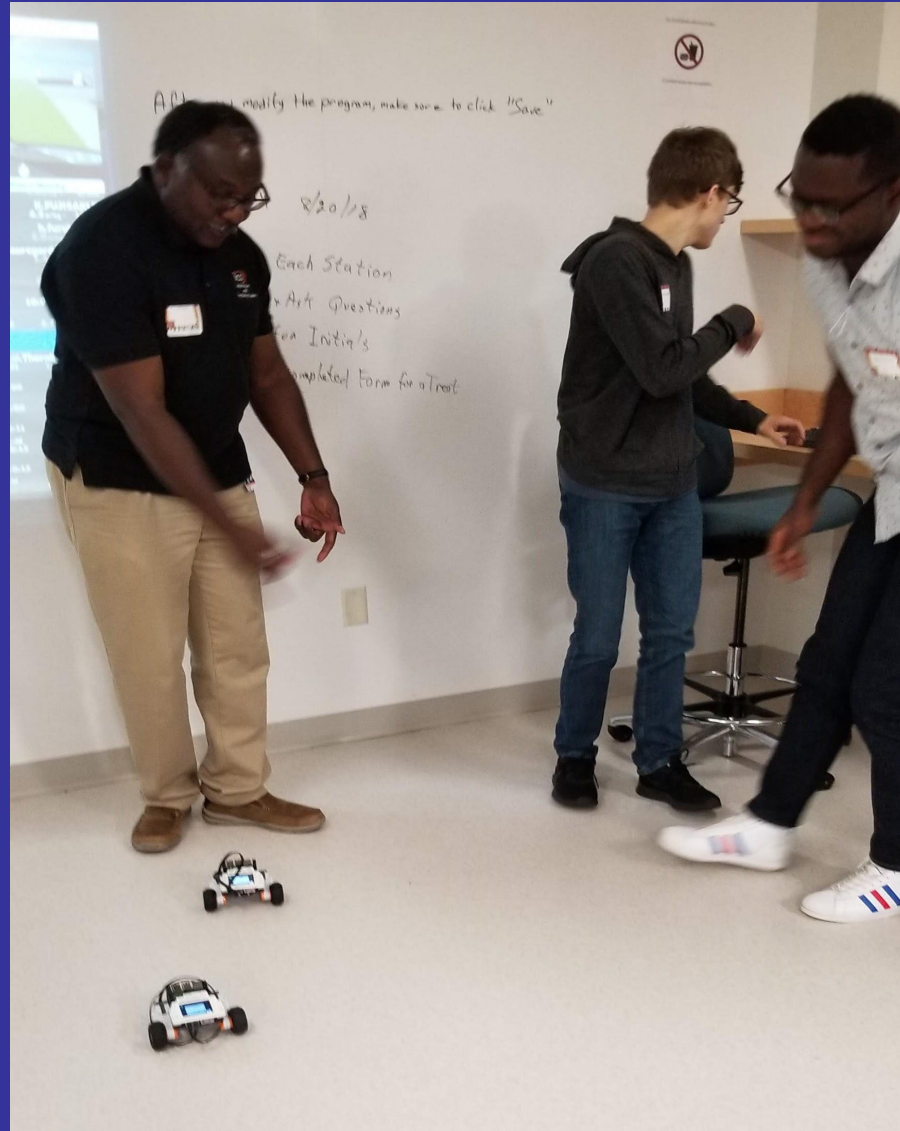
Skills Developed

- Programming low level, high level
- Hardware/software Interface - motors, etc.
- Working as a Team
- Adaptation - Building prototypes, soldering, etc.
- Time Management - deadline driven
- Presentation & Explanation Skills
    - Low level to non techie students
    - High level explaining design and programming to researchers and faculty

Question & Answer