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

RPi

Brick Pi

Pi Storm
Project Requirements

• Hardware
  – Raspberry Pi
    • Brick Pi - Lego Interface
    • Pi Storms - Lego Interface
    • Peripheral Devices (Motors, etc.)

• Software
  – Raspbian 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
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.

# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

# mindsensors.com invests time and resources providing this open source code,
# please support mindsensors.com by purchasing products from mindsensors.com!
# Learn more product option visit us @ http://www.mindsensors.com/

import os, socket
import ConfigParser
from PiStorms import PiStorms

psm = PiStorms()

config = ConfigParser.RawConfigParser()
config.read("/usr/local/mindsensors/conf/msdev.cfg")
homefolder = config.get("msdev", "homefolder")

try:
    with open(os.path.join(homefolder, ".version"), "r") as f:
        version_no = f.readline().strip()
except IOError:
    version_no = "unknown"

psm.screen.drawImage("About Me")
psm.screen.termPrintln("Device: {}".format(psm.GetDeviceId().rstrip("\0")))
psm.screen.termPrintln("Feature: {}".format(psm.psc.GetDeviceFeatures().rstrip("\0")))
psm.screen.termPrintln("f/w version: {}".format(psm.GetFirmwareVersion().rstrip("\0")))
psm.screen.termPrintln("s/w version: {}".format(version_no))
psm.screen.termPrintln("Hostname: {}".format(socket.gethostname()))
psm.screen.termPrintln("Battery: {}".format(psm.battVoltage()))

def getIP(iface):
    ip = os.popen("ifconfig {} | grep "inet addr" | cut -d: -f2 | cut -d" " -f1".format(iface)).read().rstrip()
    return ip if ip != "" else "not present"

def updateNetworkInfo():
    psm.screen.termPrintln("eth0: {}".format(getIP("eth0")))
    psm.screen.termPrintln("wlan0: {}".format(getIP("wlan0")))
    psm.untilKeyPressOrTouch(updateNetworkInfo)

psm.screen.termPrintln("Exiting to menu")
Blockly Programming

```
Logic
Loops
Math
Text
Lists
Color
Custom Vars
Functions

Motors
Sensors
Screen
LED
Buttons
System

set speed of motor BAM1 to 50
get position of BAM1
reset position of BAM1
set motor BAM1 to brake
set motor BAM1 to float
set motor BAM1 to hold
sync speed of bank A motors to 50
```
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