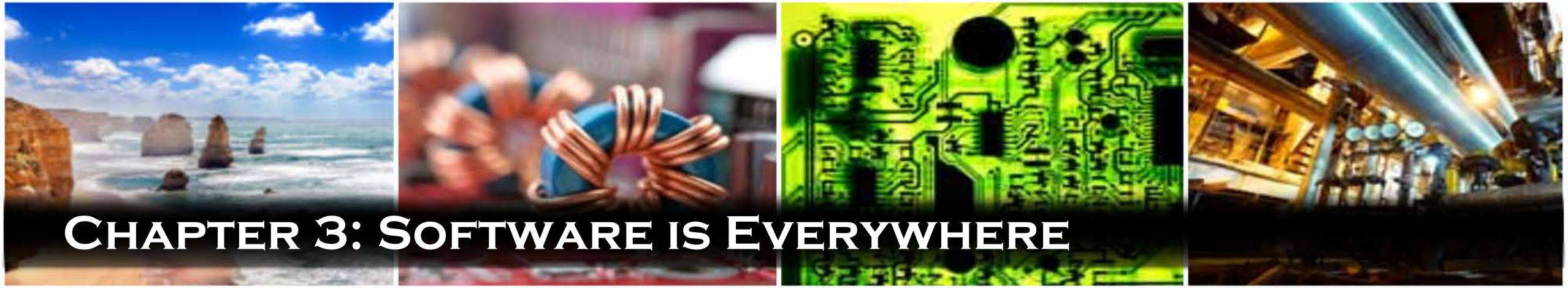


Chapter 3: Software is Everywhere

IoT Fundamentals
Connecting Things v2.0
Instructor Training

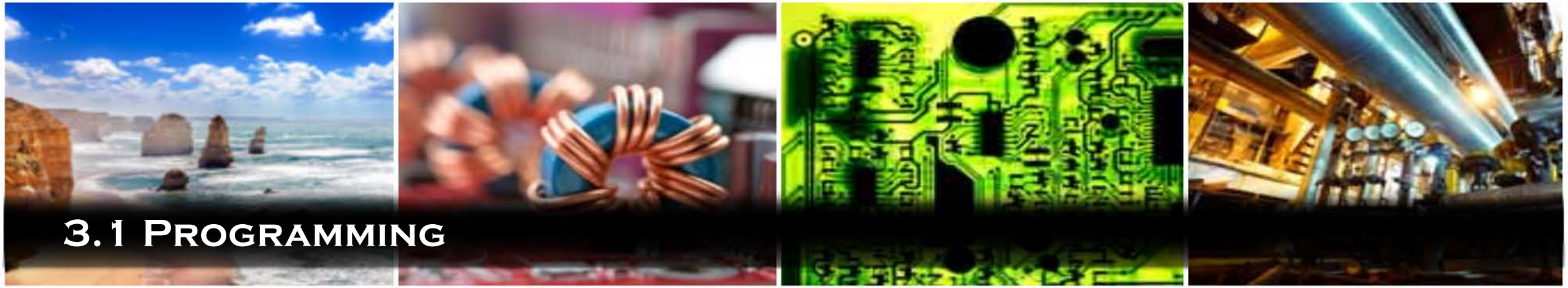
Education Service Center Region 11
Academy Support/Instructor Training Center

Chapter 3: Software is Everywhere

IoT Fundamentals
Connecting Things 2.0

Chapter 3 - Sections & Objectives

- 3.1 Programming
  - Explain the value of computer programs
- 3.2 The Raspberry Pi Single Board Computer (SBC)
  - Use the Raspberry Pi for simple applications
- 3.3 Building Models of IoT Systems in Packet Tracer
  - Use Packet Tracer to model IoT systems



3.1 Programming

Education Service Center Region 11
Academy Support/Instructor Training Center

3.1.1 What is Code?

- What is a Program?
  - A set of ordered instructions created to accomplish a specific task
  - A bread recipe can be seen as a program
  - Computer programs can be written in different programming languages
- Programs are Everywhere
  - All computers need programs
  - Operating Systems, firmware, and applications are examples of programs
- Why Learn to Code?
  - Programmers are valued in the job market
  - Today, programmers may work on firmware, device drivers, mobile applications, web interfaces, data analysis, and more
  - Programmers can create their own tools

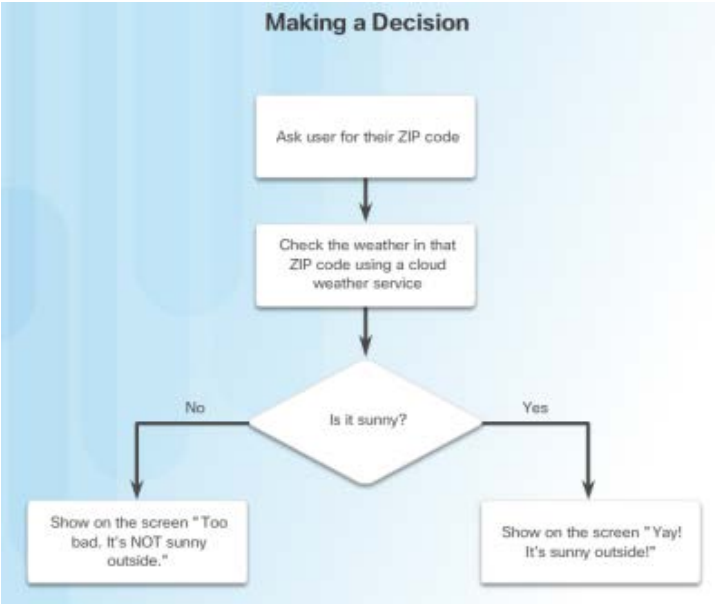
3.1.2 Code Does the Job!

- What Makes Up a Program?
  - Programs allow people impart logic to computers and are made out of logic structures
  - IF-THEN, FOR Loops, and WHILE Loops are a few logical structures commonly found in programs
- Interpreted Vs. Compiled
  - Interpreted languages rely on another program to read, parse, and execute the code
  - Compiled languages rely on a compiler, another program, to turn the human-readable code into a binary executable code
  - It is easier to maintain and troubleshoot an interpreted language
- Computer Languages
  - There are several different computer languages
  - Some computer languages are better than others at certain types of tasks
  - JavaScript, Python, Blockly, C, and Java are examples of computer languages
  - C is a compiled programming language

```
#include <stdio.h>
int main()
{
  int year;

  printf("Enter a year to check if it is a leap year\n");
  scanf("%d", &year);

  if ( year%400 == 0)
    printf("%d is a leap year.\n", year);
  else if ( year%100 == 0)
    printf("%d is not a leap year.\n", year);
  else if ( year%4 == 0 )
    printf("%d is a leap year.\n", year);
  else
    printf("%d is not a leap year.\n", year);

  return 0;
}
```

3.1.3 Lending Intelligence

- IOT Devices and Data Processing
  - A common IoT application uses sensors to collect data
  - Data is often not useful until it has been processed. Collected data is often transported and stored in the cloud for processing at a later date
- IoT Devices Make Decisions
  - Software must be written and uploaded onto IoT devices to allow them to make decisions
  - Decisions can be as simple as triggering an alarm or as complex as facial recognition
- Software APIs
  - Application Program Interface (API) is a set of routines and software tools that facilitate one application or program communicating with another
  - Different types of APIs exist: operating system APIs, application APIs, website APIs
  - APIs allow applications to communicate, share data, or ask for specific services from another application



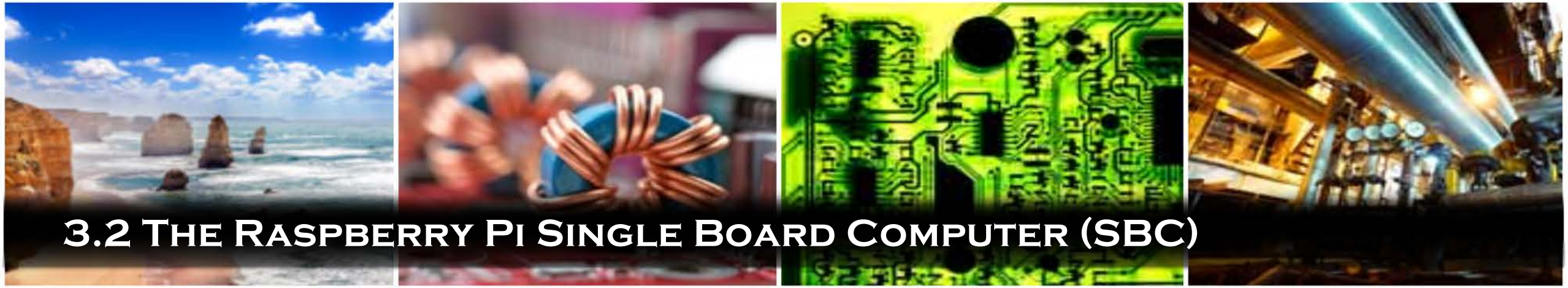
3.1.3 Lending Intelligence

- REST API
  - REST API requests trigger responses in well-defined formats such as XML or JSON
  - REST APIs use HTTP based calls between applications to access and manipulate information stored on powerful databases.
  - Web resources used to be identified using a URL. Now resources can be any entity or thing that can be addressed: today's step goal, house temperature setting, glucose setting.
  - A unique **Uniform Resource Identifier (URI)** can identify an entity
  - A URI typically begins with a slash (/steps)

GET https://www.googleapis.com/calendar/v3/calendars/***calendarID***

3.1.3 Lending Intelligence

- Securing the Code
  - Devices should protect themselves from attacks that impair its function or allow it to be used for unintended purposes without authorization.
  - Devices should protect the private authentication credentials and key material from disclosure to unauthorized parties.
  - Devices should protect the information received, transmitted, or stored locally on the device, from inappropriate disclosure to unauthorized parties.
  - Devices should protect themselves from being used as a vector to attack other devices or hosts on the Internet.
  - The **enforcement of authorized access to code** is a method to secure the control plane of an IoT device

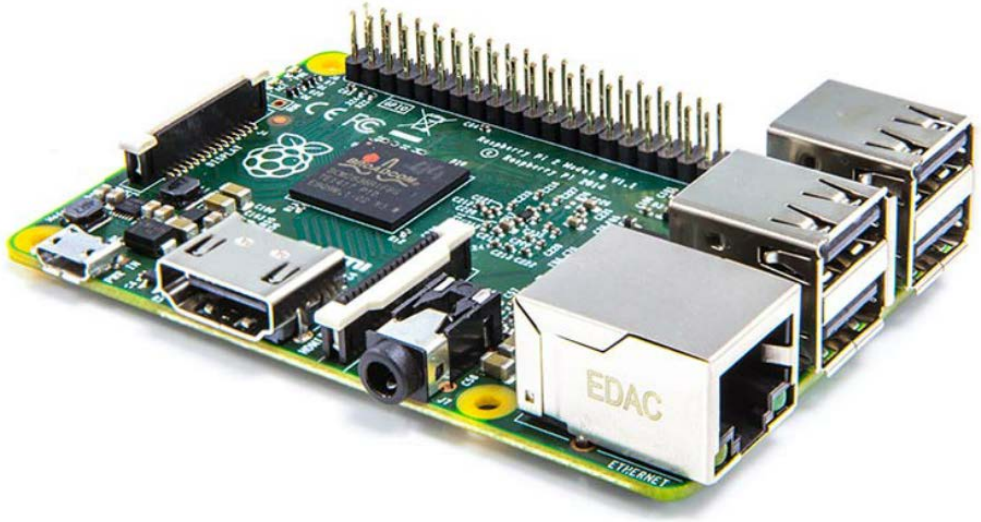


3.2 The Raspberry Pi Single Board Computer (SBC)

Education Service Center Region 11
Academy Support/Instructor Training Center

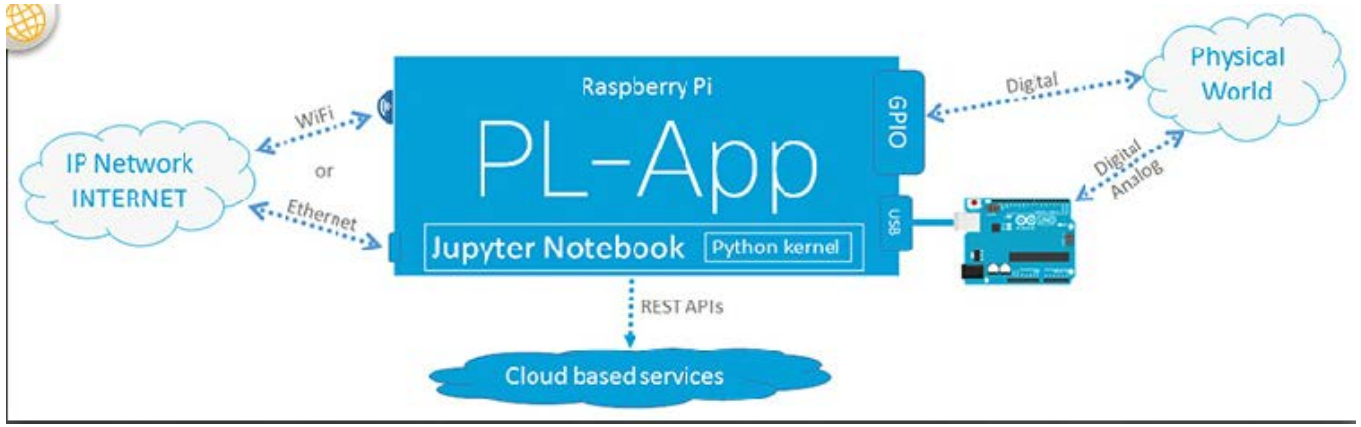
3.2.1 Raspberry Pi Hardware

- The Raspberry Pi and its Ports
  - The Pi is a small and inexpensive single-board computer
  - It has a number of USB ports that can be used to connect various devices including keyboards, mice, external drives and cameras
  - The Pi includes an 10/100Mbps Ethernet port and 40 GPIO pins, operating at 3.3V
  - Other Pi ports include an audio out, a micro SD card slot, and a micro USB (used for power) connector
- The Pi3 also adds:
  - 1.2 Ghz 64-bit quad-core ARMv8 CPU
  - 802.11n Wireless LAN
  - Bluetooth 4.1
  - Bluetooth Low Energy (BLF)
- The Pi can run a number of operating systems, including Linux and Windows



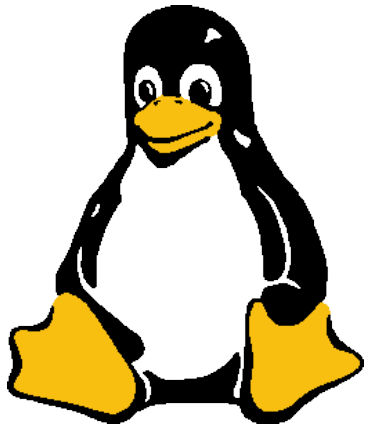
3.2.2 PL-App

- The Raspberry Pi can be accessed locally:
  - 1. Install an operating system image on the micro SD card
  - 2. Place the card in the micro SD card slot of the RaPi
  - 3. Connect a USB keyboard
  - 4. Connect a monitor or TV using the HDMI port
  - 5. Power the device with a power adapter
- The Raspberry Pi can be accessed remotely using the PL-App
- The benefit of deploying a group of headless Pi IoT devices is each can be managed remotely on a network



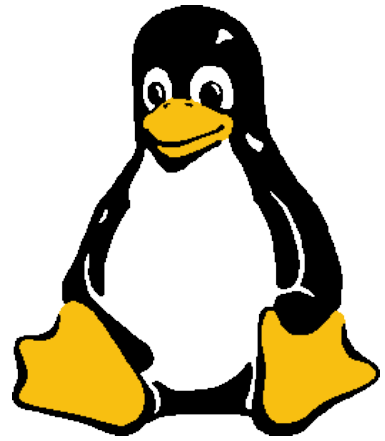
3.2.3 Using the Linux Operating System

- Understanding Linux
  - Linux is open source, fast, reliable and small and requires very little hardware resources to run
  - Supported by a community of programmers
  - Linux is part of several platforms; from wristwatches to supercomputers
  - Linux distributions include the Linux kernel, plus a number of customized tools and software packages
  - Debian, Red Hat, Ubuntu and Slackware are just a few examples of Linux distributions
  - Raspbian is a Linux distribution based on Debian and created specifically for the Raspberry Pi
- Accessing the Linux Shell
  - The Linux operating system can be divided into kernel and shell
  - The shell is a command interpreter
  - The shell is text based and also called CLI (command line interface)



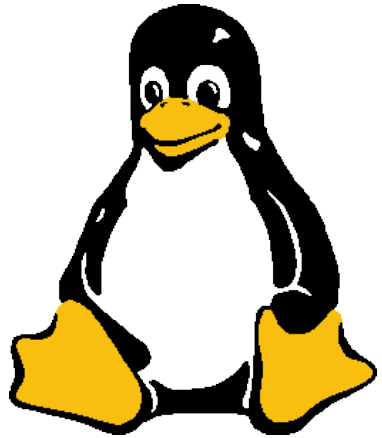
3.2.3 Using the Linux Operating System

- Accessing the CLI
  - The CLI can be accessed directly through a shell in non-graphical systems
  - Bourne Shell (**sh**), Bash (**bash**), C Shell (**csh**), improved C Shell (**tcsh**), and Z Shell (**zsh**) are popular shells
  - A terminal emulator application can be used to access the CLI in graphical environments
  - Popular terminal emulators on Linux are **Terminator**, **eterm**, **xterm**, **console**, and **gnome-terminal**



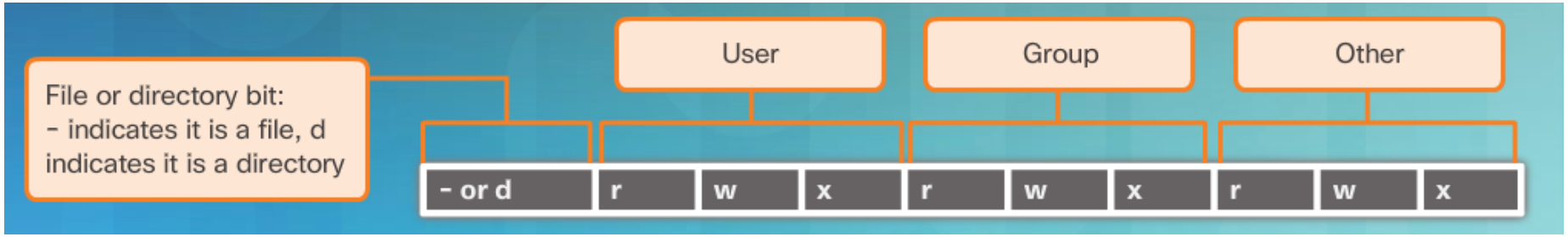
3.2.3 Using the Linux Operating System

- Basic Linux Commands
  - Linux commands are programs created to perform a specific task
  - To invoke a command via shell, simply type its name
  - Basic Linux commands:
    - **cd** - change the current directory
    - **cp** - copy a file or files from source to destination
    - **mv** - move a file or files to a different directory
    - **rm** - remove a file or files
    - **mkdir** - create a directory under the current directory
    - **grep** - search a specific string of characters within a file
    - **ps** - to list the processes currently running in the system
    - **print** - used for console output
    - **ifconfig** - used to configure, or view the configuration of, a network interface
    - **lwconfig** - used to display and change the parameters of the wireless network interface
    - **passwd** - changes passwords for user accounts
    - **pwd** - find out which directory you are currently in
  - Commands can be piped together, using the output of one as the input of the other.



3.2.3 Using the Linux Operating System

- Process Managing Commands
  - In Linux, a process is any task or command being executed by the system.
  - PIDs are unique numbers assigned to processes for identification.
  - **ps**, **top** and **kill** are commands used to manage processes.
- File Permissions
  - In Linux, most everything is treated as a file.
  - File Permissions provide a mechanism to define permissions to files.
  - Possible permissions rights are **Read**, **Write**, and **Execute** and can be defined for the user who owns the file, the group, and other system users.
  - The root user can override file permissions.

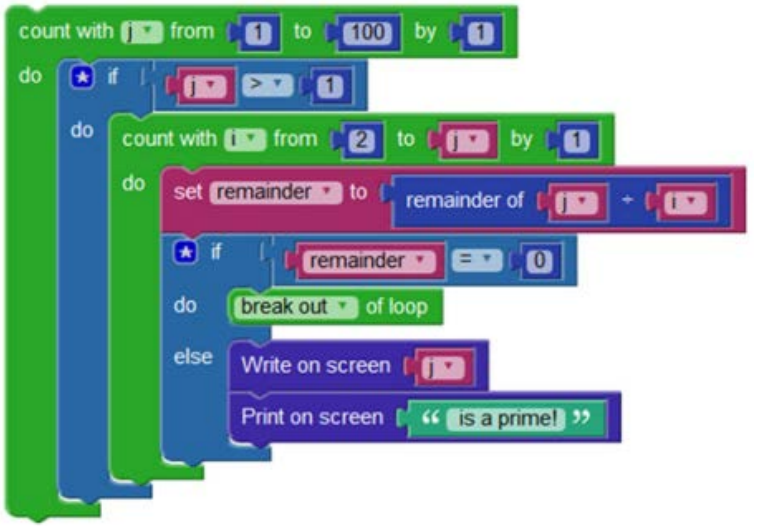


3.2.3 Using the Linux Operating System

- Package Managers
  - Maintaining computer programs and their library dependencies manually is not scalable
  - Package managers facilitate the installation, removal, and upgrade of computer programs
  - Package managers usually include user tools and a remote package repository
  - The repository hosts software packages and their dependencies
  - **dpkg** and **rpm** are popular package managers for Debian Linux and Red Hat Linux, respectively
  - Raspbian includes **dpkg** and **apt** by default
- A package manager system can be used on a Linux device to compare all installed software against a repository index, to download updates, and to install them automatically

3.2.4 Blockly

- Variables and Basic Statements
  - Blockly is a JavaScript based library for implementing visual programming
  - Blockly allows the creation of a program without entering any lines of code
  - Uses colored blocks to represent different programming structures
  - Blocks can be connected together by dragging and attaching the appropriate blocks
  - Creating a new variable in Blockly is a simple matter of dragging the variable block and filling in the value slot
- IF-THEN
  - Used to allow the code to make conditional logic decisions
- FOR Loops
  - Used to repeat the execution of a block of code for a specific number of times
- WHILE Loops
  - Used to execute a block of code while a condition is true



3.2.5 Python on the Raspberry Pi

- Using Blocky to Learn Python
  - Blockly can be used to enhance Python understanding
  - Beginners can create Blockly programs, convert them to Python and study the result
- The Python Interpreter
  - The Python interpreter understands and executes Python code
  - Python code can be created in any text editor and Python interpreters are available for many operating systems
  - Python developers can create and deploy Python programs in practically any operating system
  - When called with no arguments, the Python interpreter displays the “>>>” prompt and waits for commands; this is called **interactive mode**

```
Python 2.7 (#1, Feb 19 2010, 12:06:02)
Type "help", "copyright", "credits" or "license" for
more information.
>>>
```

3.2.3 Python on the Raspberry Pi

- Variables and Basic Statements in Python
  - Variables are labeled memory areas used to store runtime program data.
  - To assign values to variables in Python, use the = (equal to) sign.
  - Python's interactive mode implements the special variable "_".

```
>>>
>>> tax = 12.5 / 100
>>> price = 100.50
>>> price * tax
12.5625
>>> price + _
113.0625
>>> round(_, 2)
113.06
```

- Useful Functions and Data Types in Python
  - Python supports many useful functions and data types such as range(), tuples, lists, sets, and dictionary

```
list1 = ['car', 'train', 47, 2016];
list2 = [1, 2, 3, 4, 5, 6, 7 ];
print "list1[0]: ", list1[0]
print "list2[1:5]: ", list2[1:5]

When the above code is executed, it produces the following result -
 list1[0]: car
 list2[1:5]: [2, 3, 4, 5]
```

3.2.5 Python on the Raspberry Pi

- Importing Modules Into Your Code
  - Use the **import <module>** keyword to import pre-written code into your programs.
- IF THEN In Python
  - Allows the execution a block of code based on the result of an expression.
- FOR Loops in Python
  - Iterates through the items of any sequence
- WHILE Loops in Python
  - Executes a block of code while the expression is true
- Indentation is important in Python!

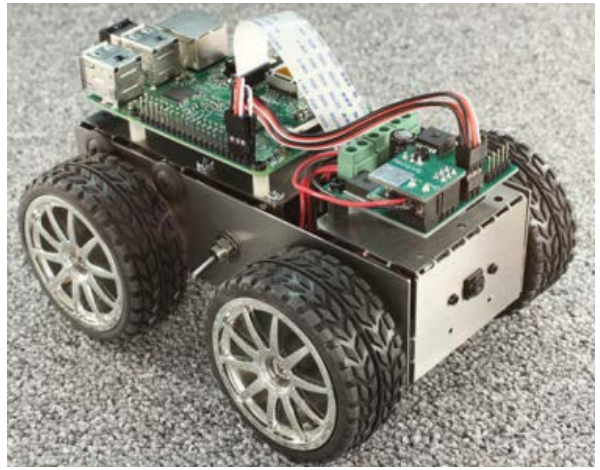
```
>>>
>>> x = int(raw_input("Please enter an integer: "))
Please enter an integer: 42
>>> if x < 0:
...     x = 0
...     print 'Negative changed to zero'
... elif x == 0:
...     print 'Zero'
... elif x == 1:
...     print 'Single'
... else:
...     print 'More'
...
More
```

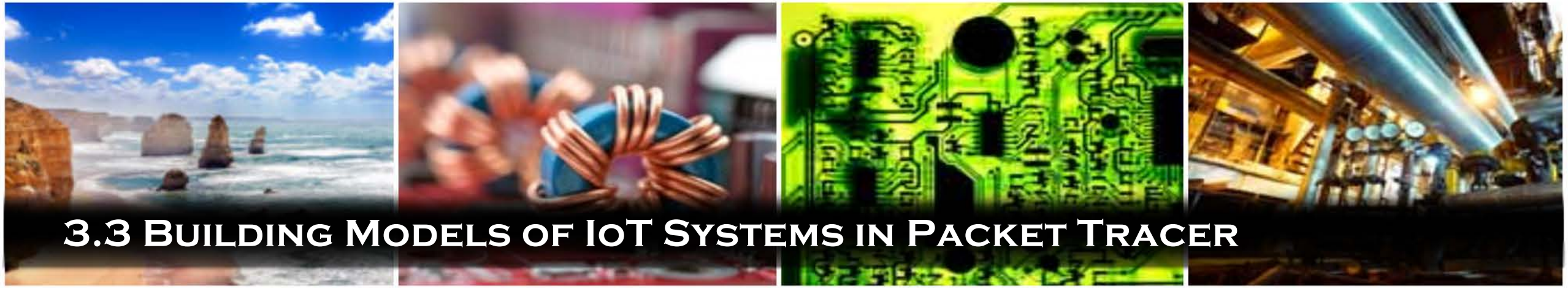
3.2.5 Python on the Raspberry Pi

- Cisco Support for Cybersecurity Professionals
  - DevNet
    - Cisco provides a beneficial community named DevNet
    - DevNet is available to assist you in learning to code, use software and programs, and partner with others
  - Cisco Spark
    - Cisco Spark is a cloud service that provides persistent chat, room-based collaboration, WebRTC video conferencing, and more
    - Developers can create code that can be used to integrate specific solutions with Spark via the Spark REST API
    - Spark REST API can include automated Spark messages based on real-world events that occur in a popular application/program

3.2.6 Uses of the Raspberry Pi

- Artificial Raspberry Pi Pancreas
  - Dana Lewis and her husband used a Raspberry Pi to build an artificial pancreas.
  - It was possible due to the Pi's small size and low power requirements
- 4Borg Pi Robot
  - PiBorg is an affordable robot kit built around a Raspberry Pi
  - It is both fun and educational
- Controlling the Arduino Through the Pi
  - While the Pi is powerful, it may not be the best option for all projects
  - The Pi doesn't include analog GPIO pins (Arduino does)
  - The Pi is **not** real-time
  - The Pi's power requirements and size may be too large, depending on the application
  - To adjust to these limitations, an Arduino may be used



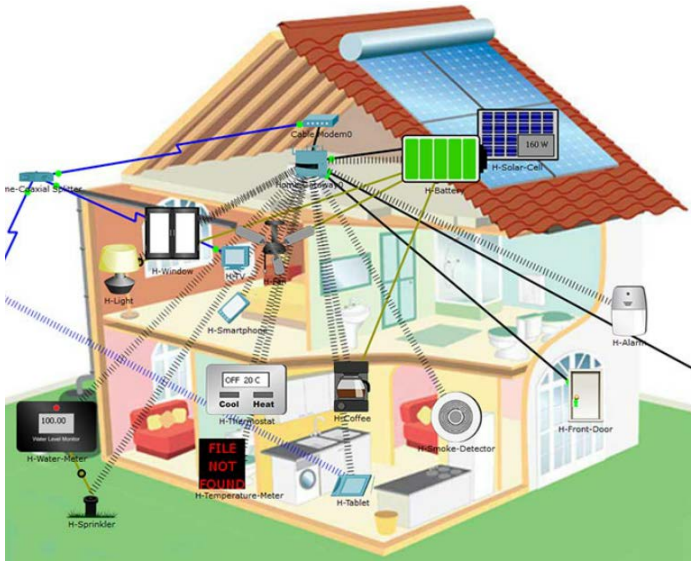


3.3 Building Models of IoT Systems in Packet Tracer

Education Service Center Region 11
Academy Support/Instructor Training Center

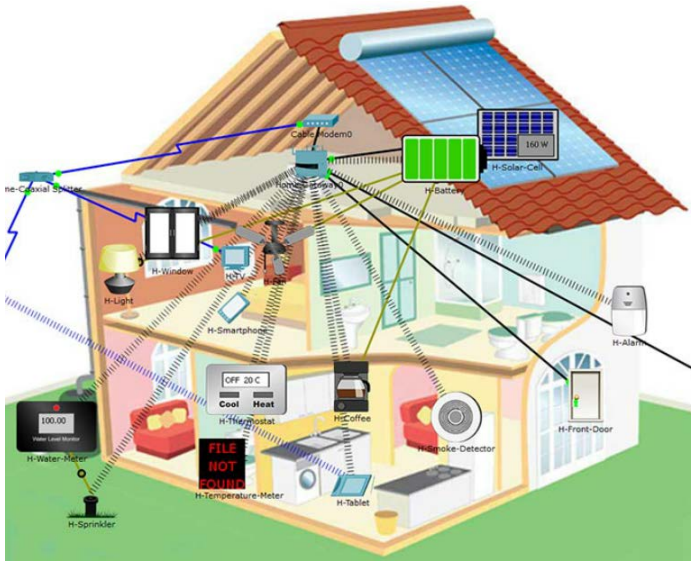
3.3.1 A Model of an IoT System

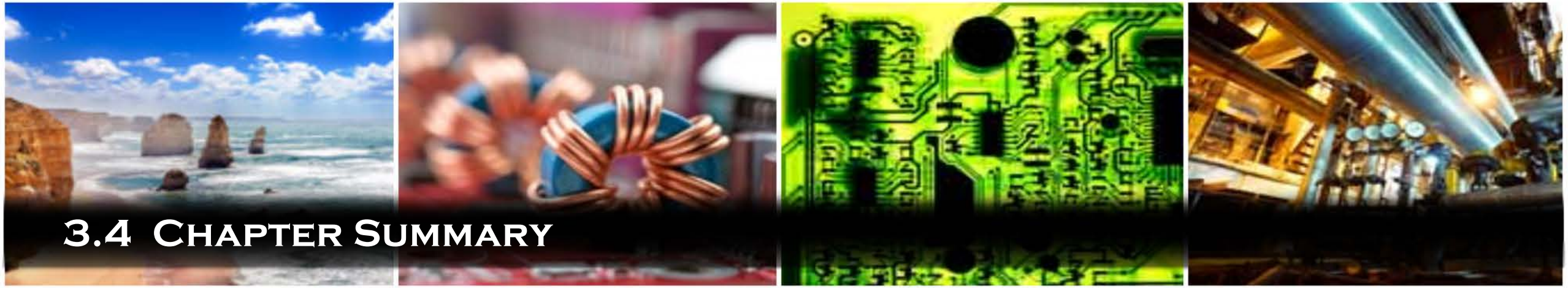
- Introducing The Home Automation Model
  - PT7.0 supports a wide range of IoT devices, such as sensors, actuators, microcontrollers, single board computers, and fog computing devices
  - PT7.0 allows the design, configuration, programming, and troubleshooting of sophisticated models of IoT systems
- The Components of the Systems
  - In the Smart Home example, all devices connect to the Home Gateway, which acts as a concentrator for all devices
  - Sensors monitor the environment while code makes sure values stay within a pre-defined threshold
  - The code also takes appropriated actions if the monitored values fall out of the pre-defined threshold
  - The cable modem and splitter pair is what provides Internet connectivity to the Home Gateway and consequently, to the entire home



3.3.1 A Model of an IoT System

- The SBC Code in Packet Tracer
  - PT 7.0 also introduces a single board computer (SBC) and a microcontroller unit (MCU)
  - PT SBC simulates an SBC such as a Raspberry Pi
  - PT SBC provides 2 USB ports and 10 digital I/O ports which can be used to connect IoT sensors and devices
  - PT SBC has a Python interpreter built in, accessible via PT SBC's Programming tab
  - PT 7.0 also supports an MCU emulator
  - PT MCU can be programmed similarly to real-word MCUs
  - PT MCU has one USB port, six digital I/O ports, and four analog I/O ports
  - PT MCU can also be programmed with Python





3.4 Chapter Summary

Summary

- Programs (also called code) are used in IoT to provide logic and intelligence to the devices. A programmer can create code to allow an IoT device to perform tasks such as monitoring, communicating to others, data processing and more.
- The Raspberry Pi, single board computer, is designed to be small and consume very little power.
- The Cisco PL-App allows access to the Raspberry Pi directly from the network without the need for a monitor, keyboard or mouse to be directly connected to the Pi.
- The Raspberry Pi runs Raspbian, a modified version of the open source and wide-spread Linux operating system.
- The Raspberry Pi supports many different programming languages including Blockly, a visual programming language, designed to help beginners learn how to program. This course focuses on Python, a popular, simple and powerful programming language.
- With added support to Python, Cisco Packet Tracer is a great tool to model, prototype and test entire IoT systems.


CISCO
EDUCATION SERVICE CENTER
REGION 11
Academy Support/Instructor Training Center