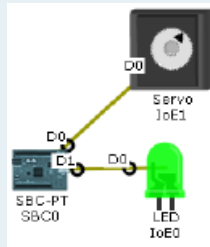


SIMULATION MIT PT7

Packet Tracer – Simulating IoT Devices

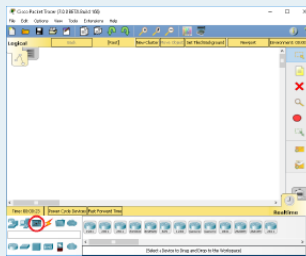
Topology



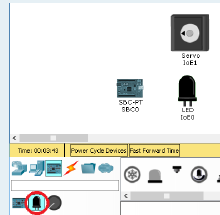
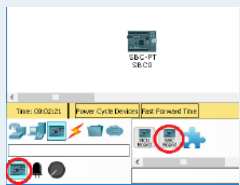
Part 1: Build the Circuit

Step 1: Place components in the logical workspace:

- ◆ Open Packet Tracer 7.0 or newer, and choose the Components icon.



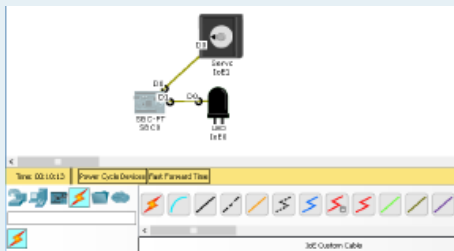
- ◆ Place a SBC Board in the Logical Workspace.



- ◆ Place an LED and a Servo in the Logical Workspace.

Step 2: Connecting the components:

- Click the Connections icon, and select an IoT Custom Cable and connect SBC0 D0 to Servo0 D0.
- Select another IoT Custom Cable and connect SBC0 D1 to LED D0.

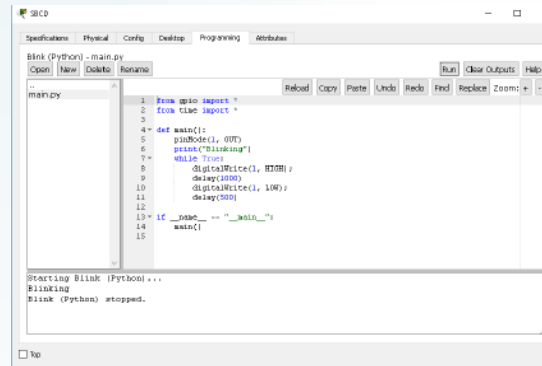


Part 2: Program the Single Board Computer (SBC)

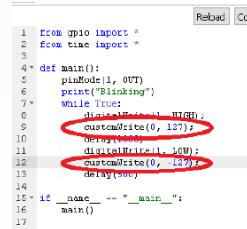
Step 1: Run the default program:

- Double-click SBC0 and select the Programming tab.
- Double-click Blink (Python) in the left pane to open it.
- Double-click main.py to reveal default Python code.
- Click the Run button to run the default code. Return to the Logical Workspace. The LED should be blinking.
- Return to the SBC0 Programming tab, and click the Stop button to stop the program execution.

Step 2: Modify the default program.



- Copy line 8 of the source code and paste it just below. Do the same with line 11 (formerly line 10).
- Modify the new lines of code to read:
customWrite(0, 127);
and
customWrite(0, -127);



- Run the modified program. The servo should now move along with the blinking LED.

Reflection:

What could be changed to make the servo turn in the opposite directions while the LED is blinking?



Join with us in
turning the perfect
storm into global
opportunities

<https://www.netacad.com/>



Cisco
IoT-Fundamentals
Connecting Things

ulrich.stritzel@itech-bs14.de
monika.stausberg@itech-bs14.de



BERUFLICHE SCHULE ITECH
Elbinsel Wilhelmsburg



IoT-Fundamentals – Connecting Things

Kursinhalt:

Nach diesem Kurs können Sie Sensoren, Aktoren, Single-Board-Computer und Cloud-Dienste über ein IP-Netzwerk verbinden und ein Ende-zu-Ende IoT System erstellen.

Vorteile:

Sie entwickeln die erforderlichen multidisziplinären Kompetenzen zur Erstellung eines Prototypen für die IoT-Lösung einer spezifischen Business-Anwendung mit Fokus auf Sicherheits-aspekte und moderne Technologien.

Lernbereiche:

- ◆ Konzepte, Möglichkeiten und Herausforderungen der digitalen Transformation durch die Verwendung von IoT verstehen und erklären.
- ◆ Sensoren/Aktoren, Mikrocontroller (Arduino), Single Board Computer (Raspberry Pi) und Cloud-Services (Cisco Spark restful API) verbinden, um ein IoT System zu erstellen.
- ◆ Relevanten Aspekte von Cybersecurity und Privatsphäre für eine IoT-Lösung verstehen.
- ◆ Verstehen, wie Digitalisierung die vertikalen Märkte verändert, wie z.B. Produktion, Energie und smarte Autos.
- ◆ Simulationstools verwenden (PT 7.0), um ein Ende-zu-Ende IoT System zu erstellen.

Zielgruppe:

Personen mit Grundlagenwissen in Netzwerktechnik und Programmierung, die Interesse haben, Digitalisierung zur Lösung gesellschaftlicher Probleme einzusetzen.

Voraussetzungen:

- ◆ Basiswissen in einer beliebigen Programmiersprache über Variablen, Operatoren, Auswahlstrukturen, Schleifen und Funktionen.
- ◆ Basiswissen zu Spannung, Strom und Widerstand und deren Zusammenhang.

Tools:

Raspberry Pi, Arduino, Prototyping Lab Cloud, Packet Tracer.

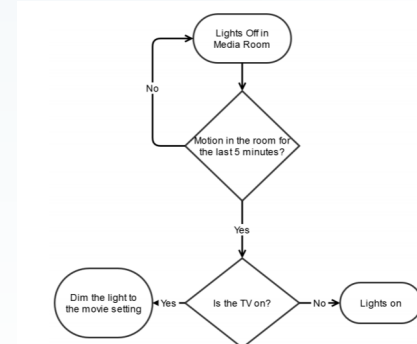
Learning Outcomes:

- ◆ Basiswissen in Prototyping-Techniken / Elektrotechnik
- ◆ Basiswissen in Python-Programmierung
- ◆ Einen Ende-zu-Ende-Prototypen entwickeln
- ◆ Eine IoT-Systemlösung entwickeln
- ◆ Ein Business-Modell für die IoT-Lösung darstellen

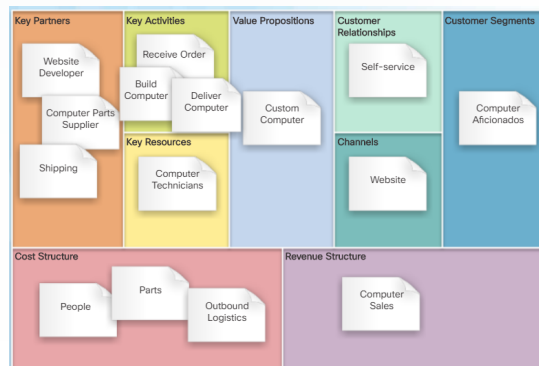
Konkrete Inhalte

1. Things and Connections	Bestandteile, Verbindungen und Informationsfluss eines IoT Systems verstehen.
2. Sensors, Actuators and Microcontrollers	Sensoren und einen Arduino benutzen, um Daten der realen Welt aufzunehmen und Aktoren zu kontrollieren.
3. Software is Everywhere	Python benutzen, um einen Single-Board-Computer (Raspberry Pi) zu programmieren und komplexe integrierte Programme zu erstellen.
4. Fog Networks and Cloud Services	Prinzipielle IoT-Netzwerkprotokolle kennen lernen. Wie werden Daten zwischen IoT System und Clouddiensten übertragen? Wie lassen sich Systeme über „restful APIs“ verbinden?
5. Industrial IoT-Applications	Wie lassen sich IoT-Technologien in vertikale Märkte bringen: Healthcare, Smart Cities, Smart Grid, Manufacturing.
6. Create an IoT-Solution	End-to-End Fallstudie zur Erstellung eines IoT-Prototypen.

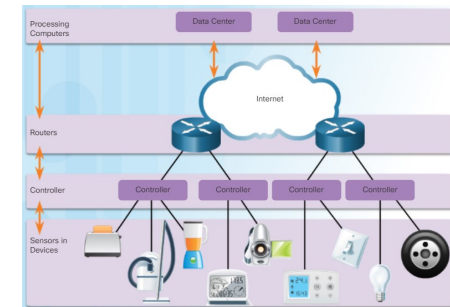
Prozesse in kontrollierten Systemen:



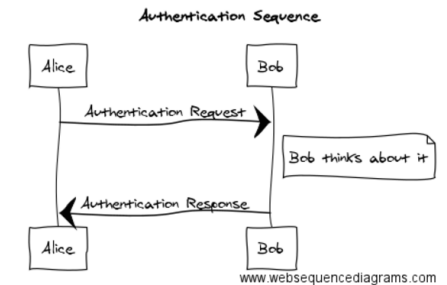
Business-Modell:



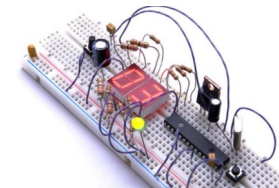
IoT-Topologie



Sequenzdiagramm:



Elektronische Schaltung:



Programmierung:

```
sketch_may09a | Arduino 1.6.8
File Edit Sketch Tools Help
sketch_may09a
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
```