Quick Start Guide: Raspberry Pi 5 ↔ Arduino Serial Communication

Get up and running in 15 minutes!

What You'll Need

- ✅ Raspberry Pi 5 with Raspberry Pi OS
- ✅ Arduino Uno R3
- ✅ USB A-to-B cable
- ✅ Internet connection

Step 1: Update System & Install Arduino IDE

Open Terminal and run these commands:

```
# Update package lists
sudo apt update

# Install Arduino IDE
sudo apt install arduino -y

# Verify installation
arduino --version
```

Expected output: `Arduino: 1.8.19` (or similar)

Step 2: Set Up USB Permissions

Add your user to the required groups:

```
# Add to dialout group (serial port access)
sudo usermod -a -G dialout $USER

# Add to tty group
sudo usermod -a -G tty $USER

# Verify groups were added
groups $USER
```

You should see `dialout` and `tty` in the output.

⚠ **IMPORTANT:** You must log out and log back in for changes to take effect!

```
# Reboot to apply changes
sudo reboot
```

Step 3: Install Python Serial Library

After rebooting, open Terminal again:

```
# Install PySerial
pip3 install pyserial
```

Or use the system package:

```
sudo apt install python3-serial -y
```

Verify installation:

```
python3 -c "import serial; print('PySerial installed successfully!')"
```

Step 4: Connect Arduino & Verify

1. **Plug Arduino into Raspberry Pi** via USB cable
2. **Arduino power LED** should light up

Check the connection:

```
# List USB devices (should show Arduino)
lsusb | grep Arduino

# Check serial port exists
ls /dev/ttyACM*
```

Expected output: `/dev/ttyACM0`

Step 5: Upload Arduino Code

Arduino Sketch (arduino_receiver.ino)

Option A: Use existing file in `RPI-Arduino_serial/arduino_receiver.ino`

Option B: Create new sketch:

1. Open Arduino IDE: `arduino &`
2. Copy this code:

```
void setup() {
  Serial.begin(9600);
  delay(2000);
  Serial.println("Arduino ready to receive!");
  Serial.println("Waiting for messages...");
  Serial.println("----------------------------");
}

void loop() {
  if (Serial.available() > 0) {
    String message = Serial.readStringUntil('\n');
    Serial.print("Received: ");
    Serial.println(message);
    Serial.println("----------------------------");
  }
}
```

3. **Select Board:** Tools → Board → Arduino Uno
4. **Select Port:** Tools → Port → /dev/ttyACM0
5. **Upload:** Click the → (Upload) button
6. **Wait for:** "Done uploading"

Step 6: Test with Python

Python Script (first_message.py)

Option A: Use existing file in `RPI-Arduino_serial/first_message.py`

```
cd RPI-Arduino_serial
python3 first_message.py
```

Option B: Create quick test script:

```
nano test_serial.py
```

Paste this code:

```
#!/usr/bin/env python3
import serial
import time

# Configuration
SERIAL_PORT = '/dev/ttyACM0'
BAUD_RATE = 9600

# Connect to Arduino
print("Connecting to Arduino...")
ser = serial.Serial(SERIAL_PORT, BAUD_RATE, timeout=1)
time.sleep(2)  # Wait for Arduino to reset

# Send message
message = "Hello from Raspberry Pi!"
print(f"Sending: {message}")
ser.write(message.encode())
ser.write(b'\n')  # Message delimiter

time.sleep(1)
ser.close()
print("Message sent! Open Arduino Serial Monitor to see it.")
```

Save (`Ctrl+O`, `Enter`, `Ctrl+X`) and run:

```
python3 test_serial.py
```

Step 7: View Results

1. **Close any running Python scripts**
2. **Open Arduino IDE** (if not already open)
3. **Open Serial Monitor:** Tools → Serial Monitor
4. **Set baud rate to 9600** (bottom right dropdown)

You should see:

```
Arduino ready to receive!
Waiting for messages...
----------------------------
Received: Hello from Raspberry Pi!
----------------------------
```

✅ **Success! Your setup is complete!**

Two-Way Communication Example

Want Arduino to respond back? Try this:

Arduino Code (with response):

```
void setup() {
  Serial.begin(9600);
  delay(2000);
}

void loop() {
  if (Serial.available() > 0) {
    String message = Serial.readStringUntil('\n');

    // Echo back with confirmation
    Serial.print("Arduino received: ");
    Serial.println(message);
  }
}
```

Python Code (with reading):

```
#!/usr/bin/env python3
import serial
import time

ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1)
time.sleep(2)

# Send message
message = "Hello Arduino!"
print(f"Sending: {message}")
ser.write(message.encode() + b'\n')

# Wait and read response
time.sleep(1)
if ser.in_waiting > 0:
    response = ser.readline().decode().strip()
    print(f"Arduino replied: {response}")

ser.close()
```

Quick Troubleshooting

"Permission denied" error

```
# Quick fix (temporary)
sudo chmod 666 /dev/ttyACM0
```

```
# Permanent fix
sudo usermod -a -G dialout $USER
# Then log out and back in
```

Port not found

```
# Check if Arduino is connected
lsusb | grep Arduino

# List all serial ports
ls /dev/tty*
```

"Port already in use"

- **Close Arduino Serial Monitor** before running Python
- **Close any Python scripts** before opening Serial Monitor
- Only ONE program can use the serial port at a time!

Arduino not responding

1. Press **Reset button** on Arduino
2. Disconnect and reconnect USB cable
3. Try a different USB port
4. Check USB cable (some are power-only)

Essential Code Templates

Standard Python Pattern

```python
import serial
import time

SERIAL_PORT = '/dev/ttyACM0'
BAUD_RATE = 9600

ser = serial.Serial(SERIAL_PORT, BAUD_RATE, timeout=1)
time.sleep(2)  # Arduino reset delay - CRITICAL!

ser.write("message".encode())
ser.write(b'\n')  # Message boundary

ser.close()
```

Standard Arduino Pattern

```
void setup() {
  Serial.begin(9600);
  delay(2000);  // Stabilization
}

void loop() {
  if (Serial.available() > 0) {
    String msg = Serial.readStringUntil('\n');
    // Process message
  }
}
```

Key Points to Remember

1. **Baud rate MUST match** in both Python and Arduino (9600)
2. **Always include delays:**
   - Python: `time.sleep(2)` after opening connection
   - Arduino: `delay(2000)` in setup()
3. **Use newline delimiters** (`\n`) to mark message boundaries
4. **Only one program** can access the serial port at a time
5. **Encode/decode text:**
   - Python sending: `.encode()`
   - Python receiving: `.decode()`

Next Steps

✅ **You're ready to start programming!**

- Try the experiment scripts: `RPI-Arduino_serial/experiment_scripts.py`
- Complete Lesson 0: `lesson_0_first_serial_message.md`
- Explore the Student User Guide: `student_user_guide.md`

Complete Setup Checklist

- ☐ Arduino IDE installed and launches
- ☐ User added to dialout and tty groups
- ☐ Rebooted after adding groups
- ☐ PySerial installed
- ☐ Arduino connects and shows in `lsusb`
- ☐ Arduino sketch uploads successfully
- ☐ Python script runs without errors
- ☐ Messages visible in Serial Monitor

Quick Command Reference

```
# Check Arduino connection
lsusb | grep Arduino
ls /dev/ttyACM*

# Launch Arduino IDE
arduino &

# Test PySerial
python3 -c "import serial; print('OK')"

# Fix permissions (temporary)
sudo chmod 666 /dev/ttyACM0

# Check your groups
groups
```

Version: 1.0 **Last Updated:** October 2025 **Estimated Time:** 15 minutes **Difficulty:** Beginner

Setup complete! Start creating! 🚀