

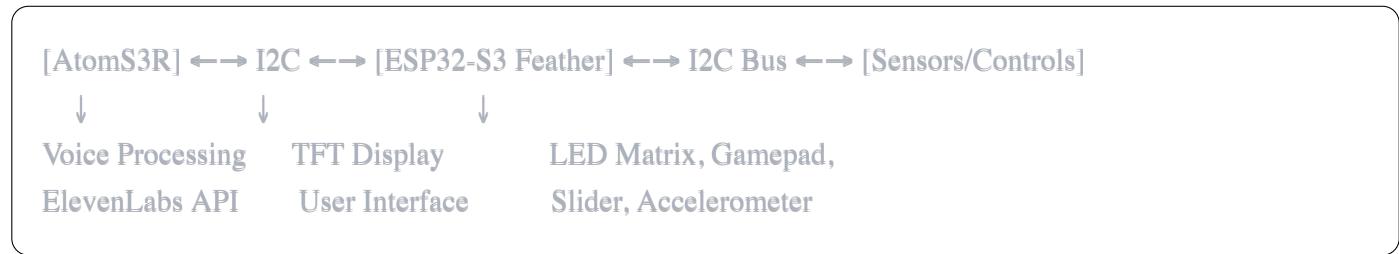
# Hardware Setup Guide - Voice Cloakroom System

## Quick Assembly Instructions for Hardware Hackathon

### Required Components

- AtomS3R AI Chatbot Kit (Voice processor)
- ESP32-S3 Reverse TFT Feather (Display controller)
- IS31FL3741 RGB LED Matrix Driver
- Mini I2C Gamepad with seesaw
- QT Slide Potentiometer with 4 NeoPixels
- LSM6DSOX Accelerometer/Gyroscope
- I2C/STEMMA QT cables (50mm and 100mm)
- USB-C cables for programming and power

### System Architecture



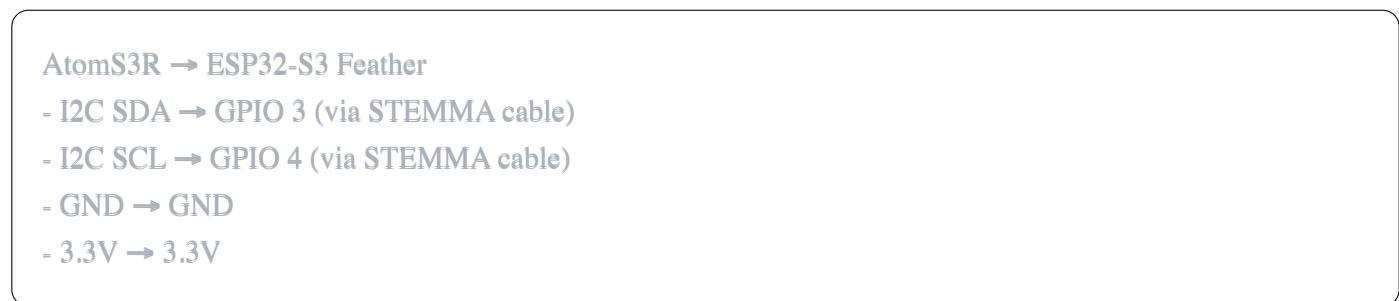
### Wiring Connections

#### I2C Bus Configuration

##### Primary I2C Bus (100kHz)

- SDA: GPIO 3 (ESP32-S3 Feather)
- SCL: GPIO 4 (ESP32-S3 Feather)
- Pull-up resistors: Built into STEMMA QT connectors

#### AtomS3R Connections



## ESP32-S3 Feather Pin Assignments

```
GPIO 3 → I2C SDA (STEMMA QT connector)
GPIO 4 → I2C SCL (STEMMA QT connector)
GPIO 7 → TFT CS
GPIO 39 → TFT DC
GPIO 40 → TFT RST
GPIO 45 → TFT Backlight
```

## I2C Device Addresses

```
AtomS3R:      0x42 (I2C slave mode)
Gamepad (seesaw): 0x50
Slider (seesaw): 0x30
LED Matrix:    0x30 (alternative address)
Accelerometer: 0x6A (default LSM6DS address)
```

## Assembly Steps

### Step 1: Power Connections

1. Connect USB-C cable to ESP32-S3 Feather for power and programming
2. Connect USB-C cable to AtomS3R for power and programming
3. Verify both devices power on (LED indicators should activate)

### Step 2: I2C Bus Setup

1. Connect STEMMA QT cable from ESP32-S3 Feather to AtomS3R
2. Use STEMMA QT hub or daisy-chain additional sensors:

```
ESP32-S3 → Gamepad → Slider → LED Matrix → Accelerometer
```

3. Ensure all connections are secure and cables are not strained

### Step 3: Display Verification

1. Upload test firmware to ESP32-S3 Feather
2. Verify TFT display shows initialization screen
3. Check LED matrix displays test pattern

4. Confirm gamepad buttons register input

#### Step 4: Voice System Test

1. Upload test firmware to AtomS3R
2. Verify microphone captures audio
3. Test speaker output for audio feedback
4. Check I2C communication between devices

### Software Deployment

#### Prerequisites

```
bash

# Install PlatformIO
pip install platformio

# Setup project
./build.sh setup
```

#### Build and Flash

```
bash

# Build all firmware
./build.sh build

# Flash AtomS3R (voice processor)
./build.sh flash atoms3

# Flash ESP32-S3 Feather (display controller)
./build.sh flash feather
```

#### Monitor and Debug

```
bash
```

```
# Monitor AtomS3R output
./build.sh monitor atoms3

# Monitor Feather output
./build.sh monitor feather
```

## System Operation

### Registration Flow

1. Press gamepad button A to enter registration mode
2. System displays "RECORDING" on TFT screen
3. LED matrix shows blue status indicator
4. User speaks keyword clearly for 3 seconds
5. System processes voice and assigns number
6. Assignment number displayed on TFT and LED matrix
7. Audio confirmation played through speaker

### Retrieval Flow

1. User approaches system and speaks keyword
2. Voice recognition processes input
3. If match found: displays assigned number
4. If no match: prompts for retry or staff assistance
5. Staff can override using gamepad controls

## Troubleshooting

### Common Issues

#### No display output:

- Check TFT connections (CS, DC, RST pins)
- Verify backlight power (GPIO 45)
- Ensure SPI bus is properly initialized

#### I2C communication errors:

- Verify SDA/SCL connections
- Check device addresses (no conflicts)
- Ensure pull-up resistors are present
- Use logic analyzer if available

### **Voice processing issues:**

- Check microphone initialization
- Verify speaker connections
- Test with known good audio samples
- Monitor serial output for error codes

### **LED matrix not working:**

- Verify I2C address (0x30)
- Check power supply (adequate current)
- Test with simple fill/clear commands
- Verify IS31FL3741 library installation

## **Error Codes**

0x01 - Voice capture failure  
0x02 - Keyword processing error  
0x03 - Storage operation failed  
0x04 - I2C communication timeout  
0x05 - Hardware initialization error

## **Performance Optimization**

### **Power Management**

- Use sleep modes between voice captures
- Adjust LED brightness based on ambient light
- Implement timeout for idle states

### **Memory Usage**

## Memory Usage

- Limit voice profile storage to 100 entries
- Use efficient data structures for I2C communication
- Implement circular buffer for audio processing

## Real-time Constraints

- Voice processing should complete within 2 seconds
- I2C communication timeout set to 100ms
- Display updates at 10Hz for smooth operation

## Hackathon Demo Script

### Demo Scenario (5 minutes)

1. **Setup** (30 seconds)
  - Power on system, show ready state
  - Explain hardware components briefly
2. **Registration** (2 minutes)
  - Register demo user with keyword "helsinki winter"
  - Show visual feedback on displays
  - Demonstrate assigned number display
3. **Retrieval** (2 minutes)
  - Same user returns and speaks keyword
  - System recognizes voice and displays number
  - Show error case with different user
4. **Features** (30 seconds)
  - Demonstrate gamepad controls
  - Show slider volume adjustment
  - Highlight LED status indicators

## Key Talking Points

- Contactless operation eliminates lost tickets

- Voice biometrics provide natural security
- Scalable for large venues (concerts, gyms)
- Works offline without internet dependency
- Handles multiple languages and accents

## Extension Ideas

### Advanced Features

- Gesture control using accelerometer
- Multi-language keyword support
- Analytics dashboard for usage patterns
- Integration with venue management systems
- Mobile app companion for staff

### Alternative Applications

- Hotel room key replacement
- Secure storage lockers
- Vehicle identification in parking
- Equipment checkout in gyms
- Laboratory sample tracking

This system transforms the traditional cloakroom experience into an engaging, technology-forward interaction while solving real operational challenges through innovative hardware integration and voice recognition technology.