# Step-by-Step Instructions: Building and Running `QuantumEdge` on Arm

## Target Platform

**Hardware**: Raspberry Pi 4 or 5 (2 GB RAM or more recommended)

**Operating System**: Raspberry Pi OS (64-bit, Bookworm)

**Architecture**: Arm Cortex-A72/A76 with NEON SIMD support

## Step 1: Prepare the Raspberry Pi

1. Install or update to **Raspberry Pi OS (64-bit)** from `https://www.raspberrypi.com/software/`.
2. If using a camera, enable it via:

   ```
   sudo raspi-config
   ```

   Navigate to `Interface Options` → `Camera` → `Enable`, then reboot.

## Step 2: Clone the Repository

```
git clone https://github.com/yourname/QuantumEdge.git
cd QuantumEdge
```

## Step 3: Install Arm-Optimized Dependencies

Run the provided setup script (executes once):

```
chmod +x setup_arm.sh
./setup_arm.sh
```

This installs:

System packages (`libatlas-base-dev`, `python3-opencv`)

ONNX Runtime for `aarch64` (Arm64) with NEON support

Required Python packages (`numpy`, `onnxruntime`, etc.)

## Step 4: Provide the Base ONNX Model

Ensure the model file exists at `models/mobilenet_cifar10.onnx`. If missing:

```
mkdir -p models
wget -O models/mobilenet_cifar10.onnx \
  https://github.com/onnx/models/raw/main/vision/classification/
     mobilenet/model/mobilenetv2-7.onnx
```

**Note**: For full originality and compliance, replace this with your own exported model generated during the Submission Period (October 6 – December 3, 2025).

## Step 5: Run the Demo

Choose one of the following modes:

### Camera Mode (Real-Time Inference)

```
python3 demo.py --mode camera
```

Press `q` to quit. Shows live feed with predicted class and latency.

### Sensor Mode (Headless Simulation)

```
python3 demo.py --mode sensor
```

Prints 10 inference steps with timing (no hardware required).

## Step 6: Verify Arm Architecture Usage

Confirm the system is using Arm-native optimizations:

```
lscpu | grep "Cortex"
python3 -c "import␣onnxruntime␣as␣ort;␣print(ort.get_device())"
```

Expected output includes `Cortex-A72` or `Cortex-A76`, and `CPU`.

## Compliance Notes

The project runs **entirely offline** after setup with no cloud or internet required.

All novel components (quantum-inspired MPS compression, on-device integration) were developed during the **Submission Period** (October 6 – December 3, 2025).

The GitHub repository is **public**, includes an **open-source license** (e.g., MIT), and contains all source code and instructions.