

# Quick Start Guide - 20 Minutes to 5-Cluster System

## Goal

Get your **5-cluster IoT routing system** with 20 sensors up and running in 20 minutes.

---

## Minute 0-4: Build Topology

### Add Devices (Drag & Drop):

1. **1 × Server-PT** (top center)
2. **1 × AccessPoint-PT** (below server)
3. **15 × MCU-PT** (5 clusters of 3 MCUs each - arrange in groups)
4. **10 × Motion Sensor** (2 per cluster, 20 total sensors)

### Connect:

- **Ethernet:** Server → AccessPoint (Copper Straight-Through)
- **IoT Custom Cable:** Each sensor → Its MCU (D0 and D1 pins)
- **Wireless:** Automatic (MCUs to AccessPoint)

### Label All Devices:

```
Server-PT BaseStation
WSN_AP
ClusterHead-1, ClusterHead-2, ClusterHead-3, ClusterHead-4, ClusterHead-5
Member1-1, Member1-2
Member2-1, Member2-2
Member3-1, Member3-2
Member4-1, Member4-2
Member5-1, Member5-2
Sensor1-1, Sensor1-2, Sensor2-1, Sensor2-2, Sensor3-1, Sensor3-2,
Sensor4-1, Sensor4-2, Sensor5-1, Sensor5-2
```

## Minute 3-6: Disable DHCP & Configure IPs

### Disable DHCP:

1. Server-PT → Services → DHCP → **OFF**

## Set Static IPs (Config Tab → Interface):

### Server:

- IP: 192.168.1.1
- Subnet: 255.255.255.0

### AccessPoint:

- IP: 192.168.1.254
- Subnet: 255.255.255.0

### MCUs (Quick Reference):

```
CH1: .100  M1-1: .101  M1-2: .102  
CH2: .110  M2-1: .111  M2-2: .112  
CH3: .120  M3-1: .121  M3-2: .122
```

All use:

- Subnet: 255.255.255.0
  - Gateway: 192.168.1.1
- 

## ⌚ Minute 6-12: Deploy Code

### For Each Device:

1. Click device → Programming tab
2. Select "UDP Socket - Python" template
3. **Delete all template code**
4. Copy corresponding code from documentation
5. Paste
6. **Save (✓)**

### Device Order (Copy codes from documentation):

1.  Server-PT (SERVER CODE)
2.  ClusterHead-1 (CLUSTER HEAD 1)
3.  ClusterHead-2 (CLUSTER HEAD 2)

4.  ClusterHead-3 (CLUSTER HEAD 3)
  5.  Member1-1 (Member1-1 code)
  6.  Member1-2 (Member1-2 code)
  7.  Member2-1 (Member2-1 code)
  8.  Member2-2 (Member2-2 code)
  9.  Member3-1 (Member3-1 code)
  10.  Member3-2 (Member3-2 code)
- 

## ⌚ Minute 12-15: Start & Verify

### Start Order:

1. **Server** → Click Run
  - Wait for: "MULTI-CLUSTER SERVER STARTED"
2. **All 3 Cluster Heads** → Click Run on each
  - Wait for: "CLUSTER HEAD X STARTED"
3. **All 6 Members** → Click Run on each
  - They start sending immediately

### Success Check (30 seconds):

#### Server Console Should Show:

```
=====
SERVER RECEIVED DATA!
From: 192.168.1.100:5001 [CLUSTER 1]
Data: CH1|M1-1:0|M1-2:1
Total Packets: 3
CH1: 1 | CH2: 1 | CH3: 1
=====
```

All Three Numbers Increasing?  SUCCESS!

---



Your 3-cluster IoT routing system is now:

- Collecting sensor data from 6 nodes
  - Aggregating at 3 cluster heads
  - Forwarding to central server
  - Displaying real-time statistics
- 

## ⚠ Quick Troubleshooting

Problem	Quick Fix
Server not receiving	Check CH1/2/3 are running and forwarding
CH not receiving	Check both members of that cluster running
Member not sending	Verify sensor connected to D0, check IP
No output at all	Click Run button, verify code saved
DHCP changing IPs	Go back and disable DHCP on server

---

## 📊 What to Demo

1. **Show Server Console** - Live data from all 3 clusters
  2. **Trigger Sensors** - Move/click them, watch values change
  3. **Simulation Mode** - Visual packet flow
  4. **Stop a Member** - Show that cluster stops forwarding
  5. **Restart Member** - Show instant recovery
- 

## 🎓 Project Features

### ⭐ Hierarchical Cluster Routing

- Energy efficient (aggregation at cluster heads)
- Scalable (easy to add more clusters)
- Fault tolerant (clusters independent)

### ⭐ Real-time Monitoring

- Live sensor data collection

- Statistical tracking
- Per-cluster analytics

## Professional Implementation

- Clean code structure
  - Error handling
  - Status monitoring
  - Packet tracking
- 

## Save Your Work

File → Save As → **YourName\_MultiCluster\_IoT(pkt)**

---

## Next Steps

### Expand Your System:

- Add more members to each cluster (M1-3, M1-4, etc.)
- Add Cluster 4, 5, 6...
- Use different sensor types (temperature, light, humidity)
- Implement data logging on server
- Add alert thresholds
- Create visualization dashboard

### Advanced Features:

- Dynamic cluster head election
  - Load balancing across clusters
  - Data encryption
  - Time synchronization
  - Energy monitoring
  - Failover mechanisms
-

## Pro Tips

1. **Always start Server first** - It needs to be listening
  2. **Label everything clearly** - Future you will thank you
  3. **Use Simulation Mode** - Great for debugging
  4. **Check IPs twice** - Most common issue
  5. **Save frequently** - Don't lose your work!
- 

## Need Help?

Refer to the complete documentation:

- **Network Architecture** - System design
  - **IP Address Scheme** - Complete IP table
  - **Code Implementation** - All 10 device codes
  - **Testing Guide** - Comprehensive troubleshooting
- 

## Final Checklist

Before demo/submission:

- All 10 devices have correct IPs
  - All codes deployed and saved
  - Server receiving from all 3 clusters
  - Packet counts balanced (CH1≈CH2≈CH3)
  - No error messages
  - Sensors triggering correctly
  - System runs continuously
  - File saved with descriptive name
  - Screenshots taken (optional)
  - Ready to present!
- 

## Congratulations!

You've built a professional-grade multi-cluster IoT routing system!

**Key Achievement:** Created a scalable, hierarchical network architecture that efficiently collects and aggregates sensor data from multiple clusters.

**Project Value:** This demonstrates real-world IoT concepts used in:

- Smart buildings
  - Environmental monitoring
  - Industrial IoT
  - Smart agriculture
  - Wireless sensor networks
- 

Time to celebrate!   