

# **Center Pivot Irrigation System - MQTT Wordpress Frontend pages Requirements**

## 1. Table of Contents

---

2.	Introduction .....	3
3.	MQTT Topic Structure.....	3
4.	Control Command Payload .....	3
5.	Status Payload .....	4
6.	Notifications Payload.....	4
7.	Device Management Commands.....	4
8.	Functional Requirements .....	5
9.	Non-Functional Requirements .....	5
10.	Access Control.....	5
11.	GUI Suggestions.....	5
12.	Mobile App GUI Mockup and Explanation.....	6
	Dashboard Page.....	6
	Control Page.....	6
	Optional Additional Pages.....	6
13.	11. Additional Control for Motor Devices.....	7
14.	Pump Motor Control (Multiple Motors Support) .....	7

## 2. Introduction

---

This document specifies the **WordPress frontend** development functional and non-functional requirements for a center pivot irrigation system that uses MQTT as the communication protocol. The system will enable remote monitoring and control of irrigation, sensor data collection, notifications, and device management (adding GPS and motor control devices).

## 3. MQTT Topic Structure

---

The system uses the following MQTT topics for communication:

- /farm/<serialNo>/control → Send control commands to the pivot.
- /farm/<serialNo>/status → Receive pivot system status updates.
- /farm/<serialNo>/sensor → Receive sensor data (soil moisture, flow rate, etc.).
- /farm/<serialNo>/notification → Receive notifications (start, stop, low pressure, alignment, etc.).
- /farm/<serialNo>/device → Send commands to add new devices (GPS, Motor).
- /farm/<serialNo>/device/status → Receive acknowledgment of device management commands.

## 4. Control Command Payload

---

Example Payload:

```
{
  "auth_key": "abc123",
  "run": true,
  "direction": "CW",
  "speed": 75,
  "mode": "WET",
  "pressure_setting": 2.5
}
```

## 5. Status Payload

Example Payload:

```
{
  "auth_key": "abc123",
  "status": "RUNNING",
  "direction": "CW",
  "speed": 75,
  "pivot_angle": 130,
  "pressure": 2.4,
  "last_tower_longitude": 44.23452,
  "last_tower_latitude": 33.32541
}
```

## 6. Notifications Payload

Examples:

```
{
  "auth_key": "abc123",
  "event": "STARTING",
  "timestamp": "2025-07-30T12:30:00Z"
}
{
  "auth_key": "abc123",
  "event": "LOW_PRESSURE",
  "current_pressure": 1.8,
  "threshold": 2.5,
  "timestamp": "2025-07-30T12:50:00Z"
}
```

## 7. Device Management Commands

To add GPS device:

```
{
  "auth_key": "abc123",
  "action": "ADD_GPS",
  "device_serial": "GPS-987654321",
  "location": { "longitude": 44.23450, "latitude": 33.32540 },
  "description": "GPS for last tower tracking"
}
```

To add Motor Control device:

```
{
  "auth_key": "abc123",
  "action": "ADD_MOTOR",
  "device_serial": "MOTOR-123456789",
  "motor_type": "PUMP",
  "description": "Pivot drive motor controller"
}
```

## 8. Functional Requirements

---

- Remote control of pivot (start/stop, direction, speed, mode, pressure setting)
- Real-time status reporting (angle, GPS location, speed, pressure)
- Collection of sensor data (soil moisture, flow rate, water used)
- Notifications for events (start, stop, alignment, low pressure)
- Device management (adding GPS and motor control devices by serial number)

## 9. Non-Functional Requirements

---

- MQTT over TLS for secure communication
- QoS 1 for control and status, QoS 0 or 1 for sensor data
- Scalability to support multiple pivot units
- Offline buffering if network is unavailable (retain message)

## 10. Access Control

---

All pages must be accessible only to the Authorized users of the farm (field authorized\_tower\_users)

## 11. GUI Suggestions

---

The Wordpress website should have at least 3 main pages:

1. Data Entry Page  
Add/edit/delete Farms and Towers data
2. Dashboard Page:
  - a. Display pivot status (Running/Stopped, Direction, Speed, Angle)
  - b. Show live sensor data (soil moisture, flow rate, pressure)
  - c. Map view with pivot location and last tower GPS position
3. Control Page:
  - a. Controls to start/stop pivot, set direction, speed, mode (Dry/Wet), and pressure.
  - b. Buttons to add new GPS or motor devices by entering serial numbers.

- c. Section to view and acknowledge notifications.

Additional pages could include a History/Logs page for past data and alerts.

## 12. Frontend Pages GUI Mockup and Explanation

The Pages for the center pivot irrigation system will have an intuitive design to help farmers easily control and monitor their pivots. The pages will be simple enough for quick access to key features but robust to handle advanced functions like device management.

### Dashboard Page

The Dashboard Page will be the main page. It will display all real-time information about the pivot system, including:

- Current status (RUNNING or STOPPED)
- Pivot direction (CW or CCW)
- Operating speed as a percentage
- Pivot angle in degrees ( $0^\circ$  = North reference)
- Live sensor readings (soil moisture, pressure, flow rate)
- A map showing the pivot layout and the GPS location of the last tower
- Notifications such as STARTING, STOPPED, ALIGNMENT, LOW PRESSURE

### Control Page

The Control Page allows the user to control the pivot system. The user can:

- Start or Stop the pivot with a single button
- Choose the direction (CW or CCW)
- Adjust the speed using a slider
- Set the irrigation mode (DRY or WET)
- Enter the desired pressure setting
- Add new devices (GPS or Motor controllers) by selecting the device type and entering the serial number.

### Optional Additional Pages

- History/Logs Page: This page will display past irrigation events, including sensor logs, water usage, and alerts in a timeline or graphical format.
- Device Management Page: This page will list all connected devices (GPS, Motors) with their serial numbers and statuses. It will allow removal or reconfiguration of devices.

This design ensures that all critical controls are accessible with minimal navigation while still providing advanced features when needed. By separating dashboard and control functions, the user experience remains clean and intuitive.

### 13. Additional Control for Motor Devices

---

In addition to controlling the entire pivot system, the app should support starting and stopping specific motor devices individually. This allows for greater flexibility in controlling different components of the irrigation system, such as pumps or drive motors.

The control command for individual motors will use the same topic ``/farm/<serialNo>/control`` but with additional parameters.

Example Payload to START a motor:

```
{
  "auth_key": "abc123",
  "action": "START_MOTOR",
  "motor_id": "MOTOR-123456789"
}
```

Example Payload to STOP a motor:

```
{
  "auth_key": "abc123",
  "action": "STOP_MOTOR",
  "motor_id": "MOTOR-123456789"
}
```

These commands will allow the backend and the pivot controller to identify which motor to operate. The ``motor_id`` corresponds to the serial number of the motor added through the device management process.

### 14. Pump Motor Control (Multiple Motors Support)

---

The system should allow control of multiple pump motors individually. Each motor will have a unique serial number and generate a Tower Unique Identifier (`tower_unique_identifier`) assigned when added to the system. The control commands will specify which motor to start or stop using its `motor_id`. This ensures that multiple pump motors can be managed independently.

Example Payloads

Start Pump Motor:

```
{
  "auth_key": "abc123",
  "action": "START_MOTOR",
  "motor_id": "PUMP-001"
}
```

Stop Pump Motor:

```
{  
  "auth_key": "abc123",  
  "action": "STOP_MOTOR",  
  "motor_id": "PUMP-001"  
}
```

If multiple motors are present, each motor will be controlled using its unique ID, e.g., PUMP-001, PUMP-002, etc.

The app's Control Page will include a section listing all available motors with individual Start and Stop buttons next to each motor. This design allows the user to control multiple pump motors independently without confusion.



## 15. Appendix

### WordPress ACF Fields

#### Farm

#	Label	Name	Field_key	Type	Remarks
1	Farm Name *	post_title	field_687fb92aa0887	Text	
2	Farm Unique ID	farm_unique_id	field_6882839af1181	Text	Generated by the system
3	Owner *	farmer_owner	field_687fb95ea0889	Select	Select from Wordpress users
4	Address 1	address_1	field_687fb974a088a	Text	
5	Address 2	address_2	field_687fb99da088b	Text	
6	City *	city	field_687fb9aaa088c	Text	
7	Province/State	provincestate	field_687fb9bfa088d	Text	
8	Country	country	field_687fbaaba088e	Select	
9	Picture 1 (Featured Image)	_thumbnail_id	field_687fbac8a088f	Image	
10	Farm Location	farm_location	field_688177229058c	Google Map	Selected on Map
11	Farm Longitude	farm_longitude	field_688385cda09ea	Number	Populated from Farm Location
12	Farm Latitude	farm_latitude	field_6883861aa09eb	Number	Populated from Farm Location
13	Authorized Users *	authorized_tower_users	field_68863fcd78b81	Select	Selected from Wordpress users

## Center Pivot Irrigation System Control Panel under Farm

#	Label	Key	Field_key	Type	Remarks
1	Tower Control Panel Serial Number *	tower_control_panel_serial_number	field_68853f1c9210d	Text	Entered manually
2	Tower Control Panel Password	tower_control_panel_password	field_68853f339210e	Text	Entered manually
3	Tower Unique Identifier	tower_unique_identifier	field_6885505aeb835	Text	Generated by the system
4	Tower Location	tower_location	field_6885507eeb836	Google Map	Selected on Map
5	Tower Latitude	tower_latitude	field_688550a9eb837	Number	
6	Tower Longitude	tower_longitude	field_688550d3eb838	Number	
7	Tower Picture	tower_picture	field_688550f3eb839	Image	
8	Tower Length (in Meters) *	tower_length_in_meters	field_688551aeeb83b	Range	
9	MQTT Application Broker Status	mqtt_application_broker_status	field_68853f509210f	Radio Button	Display from system
10	MQTT Device Broker Status	mqtt_device_broker_status	field_688551d2eb83c	Radio Button	Display from system
11	MQTT Direction	mqtt_direction	field_688551d8eb83d	Radio Button	Command
12	MQTT Agricultural Input	mqtt_agricultural_input	field_6885521feb83e	Select	Command

1 3	MQTT Preassure	mqtt_preassure	field_6885523feb83f	Number	Command
1 4	MQTT Speed	mqtt_speed	field_68855248eb840	Range	Command
1 5	MQTT Switch	mqtt_switch	field_68855267eb841	Select	Command
1 6	MQTT Tower Start Angle	mqtt_tower_start_angle	field_688653a6ad92c	Range	Command
1 7	MQTT Tower End Angle	mqtt_tower_end_angle	field_688654317690c	Range	Command
1 8	MQTT Alignment	mqtt_alignment	field_688665e7009a6	Radio Button	Command
1 9	MQTT Tower Control Panel Firmware Version	mqtt_tower_control_panel_firmware_version	field_68866250bcb47	Text	Display from system
2 0	Tower Control Panel System Type	tower_control_panel_system_type	field_68855436a6c19	Select	Entered manually
2 1	Other System Type	other_system_type	field_688554a0a6c1a	Text	Entered manually
2 2	Tower Row Number Within Farm	tower_row_number_within_farm	field_6887b96af722c	Number	Used programmatically