# Firefly Automation & Programs

The Firefly Interpreter is a scripting engine that allows you to create dynamic automation programs using your GUI panels. You can reference panel values, create custom variables, control actuators, and manage program execution flow. Programs run on the board itself, so automation works even without network connectivity.

## Basic Syntax

### Variables and Assignments

Panels act as variables that can be read and modified. You can also create custom variables in the dashboard.

```
// Reference panel values
PumpStatus = WaterPump;
if TempSensor > 30 HeatingElement = 0;


// Custom variables
TempDiff = CurrentTemp - LastTemp;
LastTemp = CurrentTemp;
```

### Operators

**Arithmetic:** + - * / ( )

**Comparison:** > (greater than), < (less than), == or is (equal to)

**Logical:** & (AND), | (OR), ! (NOT)

### Panel Integration

```
// Read panel values
CurrentTemp = TemperatureSensor;
WaterLevel = LevelGauge;


// Control actuators through panels
HeatingElement = 1;     // Turn on heating
WaterPump = 0;          // Turn off pump
```

```
FanSpeed = 75;          // Set fan to 75%


// Use timer panels
Timer1 = 300;           // Set timer to 5 minutes (300 seconds)
if Timer1 == 0 PumpProgram = 0;  // Stop program when timer expires
```

## Program Control

### One-Shot Programs

Programs that run once and shut themselves off. The last line sets the program to 0 to stop it.

```
// Pump priming program named PrimingProgram
if StartPriming WaterPump = 1;
if StartPriming pause 5;
if StartPriming WaterPump = 0;
if StartPriming PrimingProgram = 0;


// Toggle on IR remote program named IR_toggle
if IR == 13 D3 = !D3;
if IR == 13 IR_toggle = 0;
```

### Program Interaction

Programs can control other programs:

```
// Master control program
if SystemStart HeatingProgram = 1;
if SystemStart PumpProgram = 1;
if SystemStart MonitoringProgram = 1;


if SystemStop HeatingProgram = 0;
if SystemStop PumpProgram = 0;
if SystemStop MonitoringProgram = 0;
if SystemStop MasterControl = 0;
```

```
// Conditional program starting
if Temperature < 20 HeatingProgram = 1;
if Temperature > 30 HeatingProgram = 0;
```

## Practical Examples

### 1. Temperature Control System

```
// HeatingProgram
if Temperature < SetPoint - 2 Heater = 1;
if Temperature < SetPoint - 2 HeaterStatus = 1;

if Temperature > SetPoint + 1 Heater = 0;
if Temperature > SetPoint + 1 HeaterStatus = 0;

// Display current status
HeaterDisplay = HeaterStatus;
```

### 2. Water Level Management

```
// WaterLevelProgram
if WaterLevel < 20 FillPump = 1;
if WaterLevel < 20 LowWaterAlarm = 1;

if WaterLevel > 80 FillPump = 0;
if WaterLevel > 80 LowWaterAlarm = 0;

// Emergency overflow protection
if WaterLevel > 95 FillPump = 0;
if WaterLevel > 95 DrainValve = 1;
if WaterLevel > 95 OverflowAlarm = 1;
```

### 3. Timer-Based Operations

Programs can turn on Timer panels:

```
// CycleProgram

if StartCycle Timer1 = 1;

if StartCycle Pump = 1;

if StartCycle StartCycle = 0;


if Timer1 == 0 & Pump == 1 Pump = 0;

if Timer1 == 0 & Pump == 1 Timer2 = 30;


if Timer2 == 0 CycleProgram = 0;
```

### 4. Multi-Stage Process Control

```
// ProcessControl

if Stage == 1 Valve1 = 1;

if Stage == 1 & FlowSensor > 50 Stage = 2;


if Stage == 2 Valve1 = 0;

if Stage == 2 Heater = 1;

if Stage == 2 & Temperature > TargetTemp Stage = 3;


if Stage == 3 Heater = 0;

if Stage == 3 Mixer = 1;

if Stage == 3 Timer1 = 120;

if Stage == 3 & Timer1 == 0 Stage = 4;


if Stage == 4 Mixer = 0;

if Stage == 4 ProcessComplete = 1;

if Stage == 4 ProcessControl = 0;
```

### 5. Safety Monitoring

```
// SafetyMonitor
if Temperature > MaxTemp | Pressure > MaxPressure EmergencyStop = 1;
if Temperature > MaxTemp | Pressure > MaxPressure AllPrograms = 0;

if EmergencyStop Heater = 0;
if EmergencyStop Pump = 0;
if EmergencyStop AllValves = 0;
if EmergencyStop AlarmBuzzer = 1;

// Reset capability
if ResetButton & !EmergencyStop AlarmBuzzer = 0;
if ResetButton & !EmergencyStop SafetyMonitor = 0;
```

### 6. Data Logging and Monitoring

```
// DataLogger
LogCounter = LogCounter + 1;

if LogCounter >= 60 LoggedTemp = Temperature;
if LogCounter >= 60 LoggedPressure = Pressure;
if LogCounter >= 60 LoggedFlow = FlowSensor;
if LogCounter >= 60 LogCounter = 0;

// Could trigger data transmission
if LoggedTemp > 50 HighTempFlag = 1;
```

## Advanced Features

### Conditional Logic

```
// Multiple conditions
if Temperature > 25 & Humidity < 40 Humidifier = 1;
if Temperature > 25 & Humidity < 40 Fan = 0;
```

```
// Nested conditions using multiple if statements
if SystemEnabled TempCheck = 1;
if TempCheck & Temperature < 20 Heater = 1;
if TempCheck & Temperature > 30 Heater = 0;
```

### Pause Functionality

Timed sequences using pause (each statement executes in order):

```
Valve1 = 1;
pause 5;        // Wait 5 seconds
Valve1 = 0;
Valve2 = 1;
pause 10;       // Wait 10 seconds
Valve2 = 0;
SequenceProgram = 0; // End program
```

## Best Practices

1. **Use descriptive variable names**: `WaterPump` instead of `P1`
2. **Self-disable one-shot programs**: Always end with `ProgramName = 0;`
3. **Include safety checks**: Monitor for error conditions
4. **Use timers for sequences**: Better than long pause statements
5. **Document your logic**: Use meaningful panel names

## Error Handling

The interpreter will provide error messages for:

- Undefined variables
- Invalid number formats
- Division by zero
- Missing operands
- Unknown operators

Check the error output to debug your programs!

## System Variables

Two built-in variables are available in all programs:

- **time** — UTC seconds (from NTP or browser sync)
- **millis** — milliseconds since boot

## MQTT Remote Display

Home Assistant can push text to the OLED display via `cmnd/<device>/display`. The text is shown for the configured timeout. Use `\n` for an explicit line break (two lines).

---

*This interpreter integrates seamlessly with your Firefly GUI panels, making automation programming intuitive and powerful.*