Register Map

Physical Quantity	Register	Direction	Encoding
Speed Feedback	**HREG 3**	Slave to Master	12 bit count (0-4095)
Rudder Feedback	**HREG 4**	Slave to Master	12 bit count (0-4095)
Speed PWM Period (debug)	HREG 5	Slave to Master	microseconds
Rudder PWM Period (debug)	HREG 6	Slave to Master	microseconds
X Position (PLC analog simulation)	**HREG 10**	Master to Slave	stored as **meters × 100**
Y Position (PLC analog simulation)	**HREG 11**	Master to Slave	stored as **meters × 100**
Theta (Heading)	**HREG 12**	Master to Slave	**milli-radians (mrad)**

Conversion Formulas

From Modbus to Physical Values (Master side)

- Speed(m/s) - Represents a 12 bit PWM duty reading, scaled to 0–5 m/s

$$Speed\ (m/s)\ = \frac{HREG3}{4095} \times 5.0$$

- Rudder Angle (deg) - Represents a 12 bit duty reading, scaled to ±30° (will change this later)

$$Rudder(deg)\ = \frac{HREG4}{4095} \times 30$$

- X (meters) - Slave expects X stored as meters × 100.

$$X(meters)\ = \frac{HREG10}{100}$$

- Y (meters) - Same encoding as X.

$$Y(meters)\ = \frac{HREG11}{100}$$

- Theta (heading, radians)- Slave stores heading as milli-radians:

$$\theta(radians)\ = \frac{HREG12}{100}$$

Modbus Read/Write Commands Used (from Master code)

Reading Speed + Rudder (Master polling Slave)

```
mb.readHreg(SLAVE_IP, 3, &speedRaw);
mb.readHreg(SLAVE_IP, 4, &rudderRaw);
```

Writing X, Y, Theta (Master to Slave)

```
mb.writeHreg(SLAVE_IP, 10, x_reg);      // X * 100
mb.writeHreg(SLAVE_IP, 11, y_reg);      // Y * 100
mb.writeHreg(SLAVE_IP, 12, theta_reg);  // theta (mrad)
```