

CentipedeManager.h

```
typedefs
std::array<uint16_t, N_CP_PORTS>: CP_Masks

const N_CP_PORTS = 8: uint8_t
```

```
<struct> CP_Address

+port: uint8_t
+bit : uint8_t
```

```
CentipedeManager

-cp_      : Centipede
-masks_   : CP_Masks
-----
+CentipedeManager()
+begin()
+clear_masks()
+add_to_masks(CP_Address)
+set_masks(CP_Masks)
+get_masks(): CP_Masks
+report_masks(Stream &)
+send_masks()
```

constants.h

```
const NUMEL_PCS_AXIS = 15 : uint8_t
const NUMEL_LED_AXIS = 16 : uint8_t
const N_VALVES        = 112 : uint8_t

const P2VALVE[][]      : uint8_t
const P2LED[][]         : uint8_t
const VALVE2CP_PORT[]  : uint8_t
const VALVE2CP_BIT[]    : uint8_t

const N_LEDS           : uint16_t
const PIN_LED_MATRIX   : uint8_t
const PIN_R_CLICK_1    : uint8_t
const PIN_R_CLICK_2    : uint8_t
const PIN_R_CLICK_3    : uint8_t
const PIN_R_CLICK_4    : uint8_t
const R_CLICK_1_CALIB  : RT_Click_Calibration
const R_CLICK_2_CALIB  : RT_Click_Calibration
const R_CLICK_3_CALIB  : RT_Click_Calibration
const R_CLICK_4_CALIB  : RT_Click_Calibration
const DAQ_DT           : uint32_t
const DAQ_LP           : float
```

```
<struct> Omega_Calib

+balance_mA : float
+sensitivity_mA : float
+full_range_bar : float
```

```
const OMEGA_1_CALIB : Omega_Calib
const OMEGA_2_CALIB : Omega_Calib
const OMEGA_3_CALIB : Omega_Calib
const OMEGA_4_CALIB : Omega_Calib

inline mA2bar(float, Omega_Calib): float
```

ProtocolManager.h

```
typedefs
std::array<P, MAX_POINTS_PER_LINE> : Line
std::array<uint16_t, NUMEL_PCS_AXIS> : PackedLine
std::array<TimedPackedLine, MAX_LINES> : Program

const MAX_LINES = 5000 : uint16_t
(↑ make as large as free RAM allows)
const MAX_POINTS_PER_LINE =
NUMEL_PCS_AXIS * NUMEL_PCS_AXIS + 1 : uint16_t
const P_NULL_VAL = -128 : int8_t
```

```
P
("Point in the Protocol Coordinate System")

+x: int8_t
+y: int8_t
-----
+P(int8_t = P_NULL_VAL, int8_t = P_NULL_VAL)
+isNull()
+setNull()
+print(Stream &)
```

```
<struct> TimedLine

+time : uint32_t
+line : Line
```

```
<struct> TimedPackedLine

+time : uint32_t
+packed : PackedLine
```

```
ProtocolManager (work in progress)

-timed_line_buffer : TimedLine
(↑ Can/should probably be left private)

-program_ : Program
-N_program_lines_ : uint16_t
-current_pos_ : uint16_t
-----
+ProtocolManager()
+clear()
+add_line(TimedLine) : bool
+get_next_line() : TimedLine

-pack_line(Line) : PackedLine
-unpack_line(PackedLine) : Line
```

translations.h

```
VALVE2P[][] : int8_t
p2valve(P) : uint8_t
p2led(P) : uint8_t
valve2p(uint8_t) : P
init_valve2p()
valve2cp(uint8_t) : CP_Address
```