**Remote command code:**

#define IR\_CMD\_SHORT\_TEST (0x0D82)

#define IR\_CMD\_LONG\_TEST (0x0D88)

#define IR\_CMD\_PROG\_EVO\_BODY\_8W (0x0DC8)

#define IR\_CMD\_PROG\_EVO\_BODY\_16W (0x0DC9)

#define IR\_CMD\_PROG\_EVO\_BODY\_32W (0x0DCA)

#define IR\_CMD\_PROG\_EVO\_BODY\_64W (0x0DCB)

#define IR\_CMD\_PROG\_EVO\_HEAD\_4W (0x0DC0)

#define IR\_CMD\_PROG\_EVO\_HEAD\_8W (0x0DC1)

#define IR\_CMD\_PROG\_EVO\_HEAD\_16W (0x0DC2)

#define IR\_CMD\_CFG\_SD\_RESET\_TIMER (~~0x0D9F~~)

#define IR\_CMD\_CFG\_SD\_ENABLE (0x0DA0)

#define IR\_CMD\_CFG\_SD\_DISABLE (0x0DA1)

#define IR\_CMD\_CFG\_TD\_OFF (0x0DA2)

#define IR\_CMD\_CFG\_TD\_5M (0x0DA5)

#define IR\_CMD\_CFG\_TD\_10M (0x0DA4)

#define IR\_CMD\_CFG\_TD\_15M (0x0DA3)

#define IR\_CMD\_CFG\_HAS\_HEATER (0x0DB0)0DE0

#define IR\_CMD\_CFG\_NO\_HEATER (0x0DB1)0DE1

#define IR\_PROG\_SPECIAL\_FCN\_1 (0x0DE4)

#define IR\_PROG\_SPECIAL\_FCN\_2 (0x0DE5)

#define IR\_PROG\_SPECIAL\_FCN\_3 (~~0x0DE6~~)

#define IR\_PROG\_SPECIAL\_FCN\_4 (~~0x0DE7~~)

#define IR\_PROG\_ENTER\_EOL\_TEST\_MODE (0x0DF0)

#define IR\_PROG\_ENTER\_USER\_DEMO\_MODE (0x0DF2)

Extra on Remote control:

#define IR\_CMD\_COLOR\_GREEN (0x0DB0)

#define IR\_CMD\_COLOR\_RED (0x0DB1)

#define IR\_CMD\_COLOR\_BLUE (0x0DB2)

#define IR\_CMD\_COLOR\_WHITE (0x0DB3)

#define IR\_CMD\_FIRE\_ALARM\_ENABLE (0x0DD0)

#define IR\_CMD\_FIRE\_ALARM\_DISABLE (0x0DD1)

#define IR\_CMD\_FLASHER\_ENABLE (0x0DD6)

#define IR\_CMD\_FLASHER\_DISABLE (0x0DD5)

#define IR\_CMD\_REMOTE\_EXIT\_ENABLE (0x0DDD)

#define IR\_CMD\_REMOTE\_EXIT\_DISABLE (0x0DDE)

#define IR\_CMD\_REMOTE\_LAMP\_ENABLE (0x0DD9)

#define IR\_CMD\_REMOTE\_LAMP\_DISABLE (0x0DDA)

**FLASH STORED DATA:**

struct NVInformationStruct

{

    uint64\_t \_Signature;

    uint8\_t UnitWattage;

    uint8\_t HeadWattage;

    bool HasHeater;

    uint32\_t LocalLampNominalCurrent;

    uint32\_t RemoteLampNominalCurrent;

    bool SelfDiagnosticEnabled;

    uint32\_t TimeDelayPeriod; // seconds

    bool LastTestFailed;

    int32\_t HeaterTurnOnTemp;

    int32\_t HeaterTurnOffTemp;

    uint32\_t FloatChargeStartThreshold; // this is where we choose to start float charging

    uint32\_t FloatChargeFinishThreshold; // this is when we're done float charging (probably will never change)

    uint32\_t BulkChargeStartThreshold; // below this number, we'll choose to bulk charge, above it, but below the float

        // start, we'll float charge

};

{

   NVInformation.\_Signature = BackedNVInformation->\_Signature; (uint64\_t)0x2e782dade227f669

    NVInformation.LastTestFailed = BackedNVInformation->LastTestFailed;

    NVInformation.UnitWattage = BackedNVInformation->UnitWattage;

    NVInformation.HeadWattage = BackedNVInformation->HeadWattage;

    NVInformation.LocalLampNominalCurrent = BackedNVInformation->LocalLampNominalCurrent;

    NVInformation.RemoteLampNominalCurrent = BackedNVInformation->RemoteLampNominalCurrent;

    NVInformation.SelfDiagnosticEnabled = BackedNVInformation->SelfDiagnosticEnabled;

    NVInformation.TimeDelayPeriod = BackedNVInformation->TimeDelayPeriod;

    NVInformation.HasHeater = BackedNVInformation->HasHeater;

    NVInformation.HeaterTurnOnTemp = BackedNVInformation->HeaterTurnOnTemp;

    NVInformation.HeaterTurnOffTemp = BackedNVInformation->HeaterTurnOffTemp;

    NVInformation.FloatChargeStartThreshold = BackedNVInformation->FloatChargeStartThreshold;

    NVInformation.FloatChargeFinishThreshold = BackedNVInformation->FloatChargeFinishThreshold;

    NVInformation.BulkChargeStartThreshold = BackedNVInformation->BulkChargeStartThreshold;

}

**Operation Mode:**

enum class OperationModeEnum

{

    EOL\_TEST\_MODE,

    HIGH\_CHARGE\_MODE,

    FLOAT\_CHARGE\_MODE,

    HEAT\_MODE,

    EMERGENCY\_MODE,

    EMERGENCY\_FOLDBACK\_MODE, // foldsback the lamps at 45 minutes to keep battery life high

    SHORT\_TEST\_MODE,

    LONG\_TEST\_MODE,

    TIME\_DELAY\_RETURN\_MODE,

    // this will just continually charge and discharge the lamps

    DISCHARGE\_TEST\_DISCHARGE\_STANDARD\_MODE, // discharges lamps at full power

    DISCHARGE\_TEST\_DISCHARGE\_FOLDBACK\_MODE, // discharges lamps with a foldback half way through

    DISCHARGE\_TEST\_CHARGE\_MODE,

    INVALID\_MODE

};

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Requires Daughter Board | | | | | | | | |  |  | |
|  |  | Lamp Configuration | | | | | | | | |  | |
|  |  | 4 Watt | | | 8 Watt | | | 16 Watt | | |  | |
|  |  | 0 Heads | 2 Heads | 4 Heads | 0 Heads | 2 Heads | 4 Heads | 0 Heads | 2 Heads | 4 Heads | Battery Sizes | |
| Main Unit/Battery Configuration | 16 Watt | No | No | Yes | No | No | Invalid | No | Invalid | Invalid | 1 Small | |
| 32 Watt | No | No | Yes | No | No | Yes | No | No | Invalid | 1 Big | |
| 64 Watt | No | No | Yes | No | No | Yes | No | No | Yes | 2 Big | |