Luminite *Genesis* product range

Wireless PIR detectors

G1 Wireless PIR's	30m x 20m	LGWP3020
G1 Wireless PIR's	15m x 20m	LGWP1520
G1 Wireless PIR's	40m x 4.5m	LGWP4004
G1 Wireless PIR's	12m Horizontal curtain	LGWP12HC
G1 Wireless PIR's	12m Vertical curtain	LGWP12VC
G2 Wireless PIR's	30m x 20m	LG2WP3020
G2 Wireless PIR's	60m x 4m	LG2WP6004

Wireless Accessories

IP Masthead/Repeater	LGIP MT434
IP Masthead Relay Unit	LGIP MRU4x4v2
Masthead/Repeater	LGMT434v2
Masthead Relay Unit	LGMRU4x4v2
Relay Expansion Module	LGREM4x4v2
Walk Test Instrument	LGWT434
16 way relay unit	LGRU16
Relay module	LGRM8
16 way relay unit with end of line resistor	LGRU16ELR 3 versions
Relay module with end of line resistor	LGRM8ELR 3 versions
Optional antenna	AE434
Transmitter module	LGTX434
Key Point	LGKSQ
Key Fob	LGKF

Hard wired PIR detectors

G1 Wired PIR's	30m x 20m	LGHW3020
G1 Wired PIR	15m x 20m	LGHW1520
G1 Wired PIR	40m x 4.5m	LGHW4004
G1 Wired PIR	12m horizontal curtain	LGHW12HC
G1 Wired PIR	12m vertical curtain	LGHW12VC
G2 Wired PIR's	30m x 20m	LG2HW3020
G2 Wired PIR's	60m x 4m	LG2HW6004

LUMINITE ELECTRONICS LTD
2a BELLEVUE ROAD, FRIERN BARNET, LONDON, N11 3ER
Tel: 0044 (0) 208 368 7887 Fax: 0044 (0) 208 368 3952
Web: www.luminite.co.uk email: sales@luminite.co.uk

Masthead Relay Handbook

Type: LGMT434 v2

Revision1

2a BELLEVUE ROAD, FRIERN BARNET, LONDON, N11 3ER
Tel: 0044 (0) 208 368 7887 Fax: 0044 (0) 208 368 3952

PRE-INSTALLATION NOTES

Unpacking.

On receipt, inspect the package and contents for signs of damage. If damage has occurred, advise the carrier and/or suppliers immediately. Inspect the contents to confirm that all items are present and undamaged. If any items are missing or damaged, contact the supplier immediately. It is advisable that the original carton is retained as this forms the safest transport container in the event that a unit has to be returned for any reason.

Servicing.

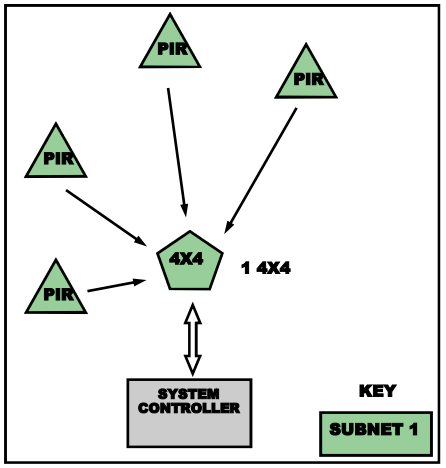
This unit should not require general servicing. Any repair work should only be undertaken by Luminite Electronics Ltd.

Moisture.

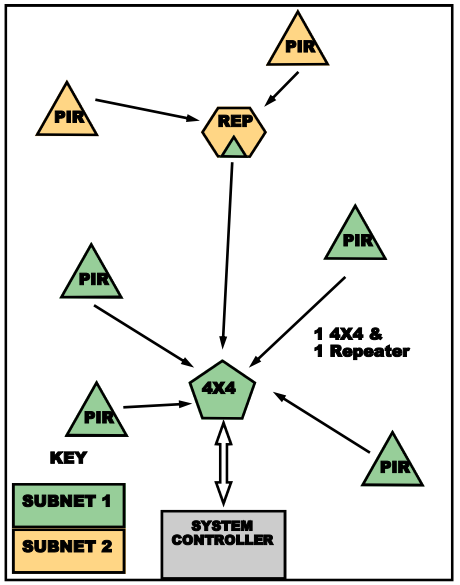
Do not expose the internal electronics of this unit to moisture i.e. take care during installation not to allow rain or damp into the product. When the product is sealed it is water resistant to IP66.

Box Contents.

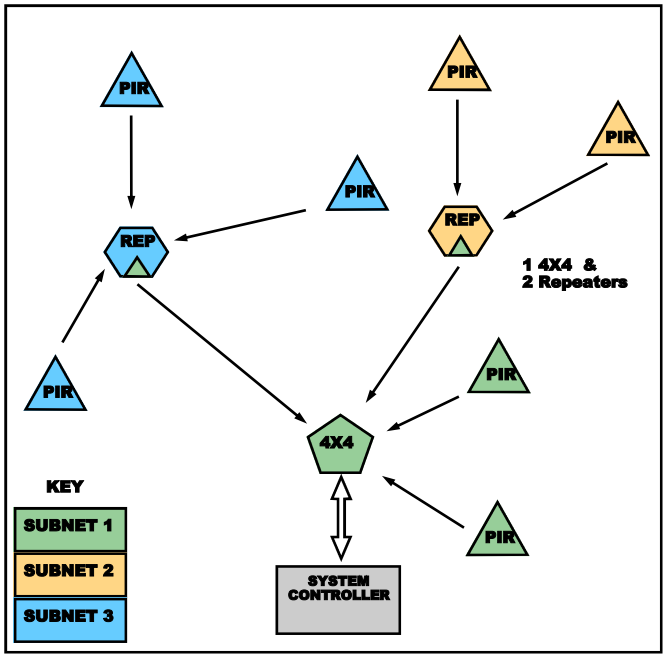
1 x *GENESIS* Masthead Relay Unit LGMT434.v2
1 x 1/4 wave antenna



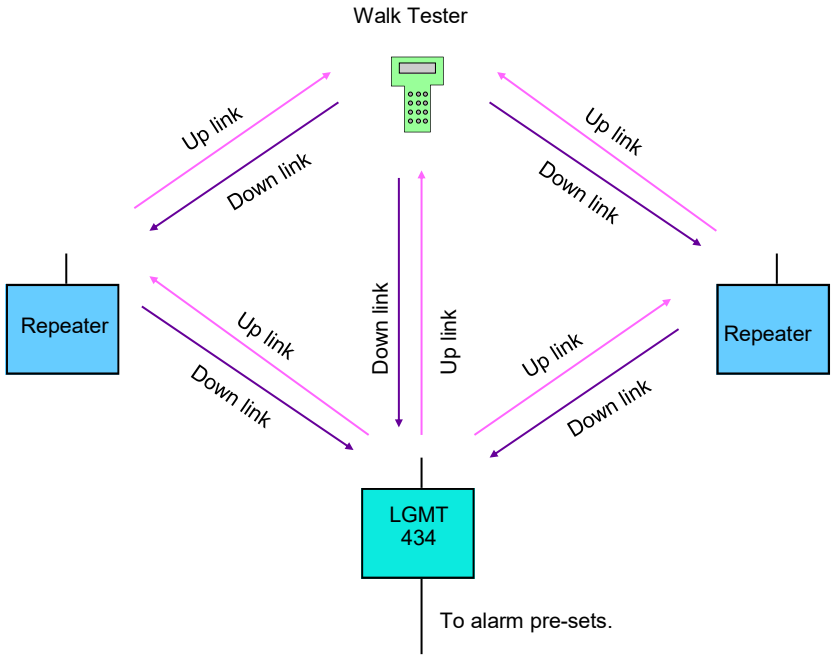
PIR's communicating directly with the LGMRU4x4v2
All Sub Nets are 1.



The other two examples show examples where one or more repeaters are used and how the Sub Net Codes separate the system.



The Walk Test Instrument does not take any notice of Sub Net codes and will receive from anywhere on the site either directly from the Masthead or via Repeaters.
All these products will be set to the same Site Code.



WALK TEST. See Test Mode on page 9. With the LGMT434v2 in test mode, when an alarm event is received it will be re-transmitted back out so that it can be received on a WALK TEST Instrument.
This is useful for testing the wireless range from the PIR detector.
Do not leave the LGMT434v2 in walk test mode as this uses a lot more air time and could cause Missing Call error messages on systems with large numbers of detectors.

Sub Net use.
Page 15 shows PIR detectors communicating directly with an LGMT434v2 and also via repeaters (LGMT434v2)
Where repeaters are used it is necessary to use more sub nets as shown.

INDEX

	page
Introduction	4
Features	5
Mode switches and connections	4
LGMRU4x4v2 Layout	6
Layout and Fitting the LGREM4x4v2 expansion module	6
Wiring	7
Programming	8
Menu options level 1	9
Menu options levels 2 & 3	10,11
Key Fob learning and default settings	12
Examples	13
Walk test	14
Sub Net use	15

Introduction.

The Genesis LGMT434.v2 Masthead Relay Unit is a receiver for Genesis wireless PIR detectors and provides simple alarm relay contacts as well as RS232 for connection to LGRU16 relay units up to 30 metres away.
Powered from 12 volts DC and weather proof to IP66 it is ideal to be fitted externally.
New features include, two volt free fully configurable relays and two auxiliary input/outputs. Remote arm/disarm is now possible with a Genesis Key Fob and/or Key Pad

Features.

- able to Learn up to 32 Keyfobs, that can control SET/UNSET state
- able to Learn up to 32 Keypoints
- able to store up to 32 PIN codes + text names, to use with Keypoints
- 2 MOS relays that can be individually configured
- 2 auxiliary I/O terminals that can be individually configured as Input with PullUp or Pull-Down, or open-drain Output (current up to 3A, 20V)
- any function (of 16 options) can be assigned to each of the 4 outputs (2 relays + 2 open-drain)
- 5 LED indicators: SIGnal, POWer, RElay1, RElay2, AUX
- SET/UNSET (ARM/DISARM) state can be controlled via Keyfobs or Keypads, or also via external input INHIBIT (AUX X2);
also can be used UART/USB command.
- several modes of operation, some work with/without Learning of PIR units, some do/don't make detection of Missing CALLs
- cooperates with Walktester
- can be used as a Repeater
- cooperates with Windows application MHManager / Analyzer and PIR-Emulator
- works with external power supply +9V to +24V connected to terminals +12V and 0V (GND),
also can be powered via USB cable (connector USB-mini, connected to PC or 5V USB charger)
- communication via RS232 or USB (virtual COM port)
- cooperates with LRGU16, including support for Optex Beams
- each PIR unit can be individually Enabled/Disabled, enabled for SET/UNSET control,
Night-only operation, and any of sensors can be disabled (Detection,Tamper,Cloak,Shock,Move.)
- all Menu parameters can be set remotely, via commands

(!!! be sure you know what you are doing)

KPPIN=12345,James Bond	saves new PIN number & assigns text_name (max.16)
KPPIN=90*5#,Paula	0-9 and also chars * # can be used in PINs
LEARNKF	starts Learning of Keyfobs
STOPLEARNKF	stops Learning of Keyfobs
LEARNKP	starts Learning of Keypoints
STOPLEARNKP	stops Learning of Keypoints
LEARNPIR	starts Learning of PIRs
STOPLEARNPIR	stops Learning of PIRs
?KFLIST .	shows list of Learned Keyfobs
?KPLIST	shows list of Learned Keypoints
?KPPINS	shows list of PINs + their text_names
?FLAGS	shows state of all RAM flags useful to find still active Tamper flags
SET_SITE=1	commands for setting of Menu parameters
SET_SUBNETA=1	sets SubnetA=1
SET_SUBNETB=0	always make SubnetB=0 to work as a Masthead !
SET_SUBNETB=2	if SubnetB is nonzero & differs from SubnetA, then works as a Repeater !
?MENU	shows listing of Menu parameters

How the system works.

At the heart of the system there is a Masthead which receives the data transmitted by the PIR detectors. See Fig 1.
Each time a PIR detects movement it transmits data to a masthead. This masthead then does two things. Firstly it re-transmits the data and attaches a text string to it which can be received by either a Pager (LGP434) or Walk Test Instrument LGWT434). These devices will display the ID number and name of the PIR such as “Main Gate” etc.
The walk test unit shows additional information such as the PIR’s battery status, pulse count setting and software version as well as the signal strength at the masthead.

The second thing that the masthead does is to pass data via an RS232 link to an alarm system such as the LGRU64 relay control unit with up to 64 alarm relay outputs and a further 64 tamper relay outputs. This control unit keeps a log of events and provides text for video insertion.
Alternatively, the masthead may be connected directly to some popular DVR/transmission systems thus illuminating complex and costly wiring.
Transmission from the detector to the masthead transceiver can be up to 1km line of sight but may be extended to many kilometres by using repeaters.

A masthead unit can simply be converted to become a repeater by just a switch change.
The PIR detectors CALL IN every 5 minutes with an IM’E STILL HERE message.
The LGRU64 and some third party products will give an alert if any of the PIR’s logged on the system fail to call in.

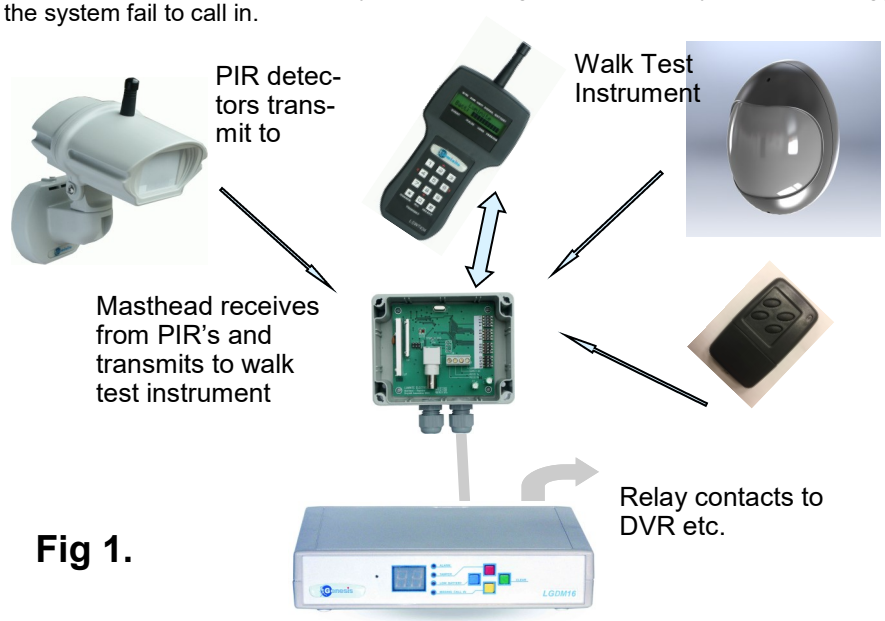
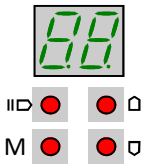


Fig 1.

PROGRAMMING:

The new version LGMT434v2 now has an LED display and push buttons to set the mode of operation.
This makes changing settings much easier and facilitates more options and features.

Button functions.



SELECT UP
M
MEMORISE DOWN

Power On,
When the power is turned on, the display shows momentarily.

How to set standard functions.
Step 1. Enter the settings mode. Press **M** until the display shows

Step 2. Scroll through the mode options with this button EG:

When the desired mode is reached, wait a moment and then the display will flash the setting. EG: = site code 1.

To change a setting, select the mode and wait for the setting to flash.
Scroll up or down to the desired number.
Step 3. Press and hold the M button to store the new number.

The display shows followed by

Now let go of the button and the new setting is now stored.

EXAMPLE: **M** **M**

To change more than one setting at a time, follow steps 1 and 2 and after changing the setting press to select the next mode. Press the **M** button when all changes have been made.

NB: The display will automatically turn off after 10 minutes if M is not pressed.

List of implemented commands
Enter command **?HELP**
HELLO ..... answers ACK, to check Masthead communicates
RESET ..... restarts, loses USB connection
?INFO ..... shows info about current HW and firmware
CALIBRATE ..... starts RF calibration
DOTXE ..... makes Easyswitch transmission test
DOTXG ..... makes Genesis transmission test (Site1/Sub1/PIR1/D-- S-)
PIRTEXT=1,NUMBER 1 ..... assigns text_name [NUMBER 1] to PIR1
?PIRSTAT ..... shows text_names & individual settings of all PIRs
?PIRSTAT=ALL ..... the same
?PIRSTAT=LE ..... the same, but only Learned PIRs
?PIRSTAT=1 ..... the same, but only selected PIR unit
PIRCFG=XXX,Y,SSS ..... individual configuration of selected PIR unit
XXX: units to configure, 1..64 or ALL
Y: E=Enable / D=Disable / S=Set
SSS: combination of switches:
D = Detect (pyro)
T = Tamper (lid switch)
C = Cloak (IR-antimasking)
S = Shock (piezo)
F = Fault (reserved)
A = Accelerometer Move, position change
M = Missing (timeout)
V = Violation (code violation)
E = Enabled individually
L = Learned
U = Under SET/UNSET control, individually
N = Night only operation
PIRCFG=1,E,TCS ..... PIR_1, Enable: Tamper Cloak Shock
PIRCFG=27,D,S ..... PIR_27, Disable: Shock
PIRCFG=ALL,S,DTCSFAMVEU .... all PIRs, Set: list of switches
PIRCFG=DEFAULT ..... like the previous command, this is default
ERASEPIRNAMES ..... erases all text_names assigned to PIRs
ERASEPINS ..... erases all Keypoint PINs and assigned names
ERASEKP ..... erases all Learned Keypoints and RollingCodes
ERASEKF ..... erases all Learned Keyfobs
ERASEMENUPARS ..... erases all Settings area controlled via Menu
then default values are written
ERASEOTHERPARS ..... erases Other settings, controlled only via commands
MAKEBRANDNEW ..... gets device to state like brand new board
(!!! clears all memories & sets default settings)
PRINTI2C ..... shows content of all I2C-EEPROM memory
ERASEI2C ..... erases I2C-EEPROM (RollCodes, BattLevels, RSSI)
PRINTFLASH ..... shows content of important SPI-Flash sections
WRITEFLASH=4095,128 ..... writes decimal number to SPI-Flash
WRITEFLASH=$FFF,$80 ..... the same with hexadecimal numbers

stoplearnkp *// entered command STOPLEARNKP*
ACK
Learning Keypoints done *// now is in normal operation mode*
KPUNSET 12345 PIN[0] KP[0] *// Keypoint was used, 12345 UNSET*
[KP_U_U] *// Keypoint was used, wanted_UNSET, made_UNSET*
[UNSET] *// state has changed (was SET before)*

How to check if any Keypoint was Learned or how many:
Enter command **?KPLIST**
?kplist *// entered command ?KPLIST*
ACK
Learned Keypoints: *// header*
00: 00A6EDC7 *// first item (index [0] and 32b DeviceID)*
total 1 Keypoints *// total count*

How to Erase all Learned Keypoints:
Enter command **ERASEKP**
It Clears (Deletes) all Keypoints at once.
erasekp *// entered command*
ACK
erased all Learned Keypoints
and RollingCodes *// shows confirmation*

Menu Parameters

command to set it:
===== Menu Level 1 [n1] ====== (Basic Settings)

"SI" ... Site, 1..32 {def.1} SET_SITE
"SU" ... Subnet (A), 1..8 {def.1} SET_SUBNETA
"Sb" ... Subnet_B, 0..8 {def.0} SET_SUBNETB
"LM" ... Learning_Mode, 0..3 {def.0} SET_LEARNMODE
0 .. no Learning, no Missing detection, legacy mode for LGRU16
1 .. no Learning, makes Missing detection for PIRs 1x received since power-up
2 .. needs Learning, ignores not-Learned PIRs, no Missing detection
3 .. needs Learning, ignores not-Learned PIRs, makes Missing detection
"tA" ... Time ALARM, 0..99 {def.2} SET_TIMEA
"tt" ... Time TAMPER, 0..99 {def.2} SET_TIMET
lengths of relay activity
value 0 ..... 250ms
values 1-60 ... 1-60s
values 61-99 .. (X-60) minutes (examples: 63 => 3 min, 99 => 39 min)
"r1" ... RELAY_1 Idle state, 0=N/O=Off, 1=N/C=On {def.0} SET_IDLERE1
"r2" ... RELAY_2 Idle state, 0=N/O=Off, 1=N/C=On {def.0} SET_IDLERE2
"r3" ... AUX_X1 Idle state, 0=N/O=Off, 1=N/C=On {def.0} SET_IDLERE3
"r4" ... AUX_X2 Idle state, 0=N/O=Off, 1=N/C=On {def.0} SET_IDLERE4
"LC" ... Light Control, 0..64 {def.0} SET_LIGHTCTRL
0 ..... Light Control Disabled;
Detections work no matter if Day or Night
1-64 .. Light Control Enabled;
number is PIR Unit that gives the Day/Night information,
Detections work only on Night (also Masthead must be in SET state)
"tM" ... Test Mode, 0=Disabled, 1=Walktest_Mode_Enabled {def.0}
SET_TESTMODE
===== Menu Level 2 [n2] ======= (Configuration of Tamper Relays)

"tP" ... Tamper, 0=Disabled, 1=Enabled {def.1} SET_ENTAMPER
"CL" ... Cloak, 0=Disabled, 1=Enabled {def.1} SET_ENCLOAK
"SH" ... Shock, 0=Disabled, 1=Enabled {def.1} SET_ENSHOCK
"FA" ... Fault, 0=Disabled, 1=Enabled {def.1} SET_ENFAULT
"MO" ... Move, 0=Disabled, 1=Enabled {def.1} SET_ENMOVE
"MI" ... Missing, 0=Disabled, 1=Enabled {def.1} SET_ENMISS
"bL" ... Battery_Low, 0=Disabled, 1=Enabled {def.1} SET_ENBATTLOW
"cu" ... Code_Violation, 0=Disabled, 1=Enabled {def.0} SET_ENCODEVIO
"Ld" ... Lid_Opened, 0=Disabled, 1=Enabled {def.0} SET_ENLIDOPEN

===== Menu Level 3 [n3] ======= (Configuration of Global Relays)

"F1" ... Function_1 (RELAY_1), 0..14 {def.1} SET_FNC1
"S1" ... Source/Start Index for Function_1, 1..64 {def.1} SET_SEL1
"E1" ... End Index for Function_1, 1..64 {def.64} SET_END1
"F2" ... Function_2 (RELAY_2), 0..14 {def.2} SET_FNC2
"S2" ... Source/Start Index for Function_2, 1..64 {def.1} SET_SEL2
"E2" ... End Index for Function_2, 1..64 {def.64} SET_END2
"F3" ... Function_3 (AUX_X1), 0..16 {def.9} SET_FNC3
"S3" ... Source/Start Index for Function_3, 1..64 {def.1} SET_SEL3
"E3" ... End Index for Function_3, 1..64 {def.64} SET_END3
"F4" ... Function_4 (AUX_X2), 0..16 {def.15} SET_FNC4
"S4" ... Source/Start Index for Function_4, 1..64 {def.1} SET_SEL4
"E4" ... End Index for Function_4, 1..64 {def.64} SET_END4
"AC" ... ARM/DISARM Control mode, 0..8 {def.7} SET_ARMCTRL
0 .. no ARM/DISARM control (ALARM relays always work on Detections)
1 .. ARM/DISARM control enabled, via Keyfobs
2 .. ARM/DISARM control enabled, via Keypoints
3 .. ARM/DISARM control enabled, via Keyfobs & Keypoints
4 .. ARM/DISARM control enabled, via input INHIBIT (INHIBIT has priority)
5 .. ARM/DISARM control enabled, via input INHIBIT & Keyfobs
6 .. ARM/DISARM control enabled, via input INHIBIT & Keypoints
7 .. ARM/DISARM control enabled, via input INHIBIT & Keyfobs & Keypoints
8 .. ARM/DISARM control enabled, only via UART/USB command
Note: SET/UNSET commands sent via Keyfobs & Keypoints may optionally wait for confirmation (UART/USB command) from Alarm system.
"SC" ... Start Index for ARM/DISARM Control, 1..64 {def.1} SET_STACTRL
"EC" ... End Index for ARM/DISARM Control, 1..64 {def.64} SET_ENDCTRL
"rS" ... Restore Mode, 0..1 {def.1} SET_RESTORE
0 .. restart with all relays Idle and Keyfob_state = SET (and KC_state Off)
1 .. restart with last known states (reload from EEPROM on power-up)
"EM" ... Extra Mode, 1..3 {def.1} SET_EXMODE
1 .. print standard CALL & ALARM messages (Legacy mode)
2 .. print „Short texts“, (Control Unit mode)
3 .. print both, standard CALL & ALARM messages + „Short texts“ (debugging)
===== Menu Level 4 [n4] ==== (Commands for Learning Keyfobs & Reset)

"LE" ... Learn Keyfob 1 = do it {def.0}
"Er" ... Erase ALL Learned Keyfobs 1 = do it {def.0}
"dS" ... Default Settings 1 = do it {def.0}
==========

Keypoints

Before a Keypoint can be Learned, user must define at least one valid PIN code to use.
Use command KPPIN to store a PIN to nonvolatile memory.
PIN codes have fixed length 5 characters, can consist of decimal digits 0..9 and chars * # .
A text name is assigned to each PIN code, it should ease identification of who made a particular SET/UNSET operation. Text name can be max. 16 chars long, is not case sensitive (all characters will change to capitals), can contain any letters, numbers, spaces, underscores,
chars +-*/!?@#$%^&<>(){},.:; (all ASCII chars range 0x20 .. 0x7E, excluding [ ]).

kppin=12345,James Bond // saves new PIN code 12345 & assigns name
ACK
added PIN #1: 12345 [JAMES BOND]
*kppin=90*5#,User{37}-Peter // PIN can consist of 0..9 and chars * #*
ACK
*added PIN #2: 90*5# [USER{37}-PETER]*

How to check if any PINs were defined or how many:
Enter command **?KPPINS**
?kppins // entered command ?KPPINS
ACK
Keypoint PINs: // header
00: 12345 [JAMES BOND] // first item (index [0], the PIN code and text_name)
total 1 Keypoint PINs // total count
How to Erase all PINs for Keyfobs:
Enter command **ERASEPINS**
It Clears (Deletes) all PIN codes + their TextNames, all at once.
erasepins // entered command ERASEPINS
ACK
erased all Keypoint PINs and assigned names // shows confirmation
How to Learn a new Keypoint:
User must switch to „Learning of Keypoints“,
Enter command **LEARNKP**
Then use the Keypoint, enter any of valid PIN codes (stored before),
+ choose any operation SET/INFO/UNSET; the Keypoint will be Learned (its 32b DeviceID).
Stop „Learning of Keypoints“ via command **STOPLEARNKP** , or make power cycle.
learnkp // entered command LEARNKP
ACK
Learning Keypoints..
Keypoint - new DeviceID record created // printed after Keypoint use: 12345 INFO
KPINFO 12345 PIN[0] KP[0] new // used pin [0], new Keypoint stored at memory [0]
[KP_I_S] // Keypoint was used, wanted_INFO, state_is_SET

How to check if any Keyfob was Learned or how many:

Enter command **?KFLIST**

?kflist *// entered command ?KFLIST*
ACK
Learned Keyfobs: *// header*
00: 279C78B6 *// first item (index [0] and 32b SerialNumber)*
total 1 Keyfobs *// total count*

How to Erase all Learned Keyfobs:

Enter command **ERASEKF**

erasekf *// entered command ERASEKF*
ACK
erased all Learned Keyfobs *// shows confirmation*

Notes:
Up to 32 Keyfobs can be Learned.
Deleted can be only all at once.
When SET button on Keyfob is pressed (top-left), a red LED lids continuously (Keyfob makes transmissions repeatedly). When Masthead receives the command from Keyfob, it makes the required operation (turns to SET state) and then transmits confirmation (feedback) to the Keyfob.
The LED starts flashing after confirmation received.
Similar applies to UNSET button (top-right) that lids green.

User must hold SET or UNSET button on Keyfob until LED changes from continuous light to flashing. If button released sooner, then the state change is not guaranteed.

Menu Parameters

===== Menu Level 1 [n1] ====== (Basic Settings) ==========command to set it:

"SI" ... Site,	1..32 {def.1}	SET_SITE
"SU" ... Subnet (A),	1..8 {def.1}	SET_SUBNETA
"Sb" ... Subnet_B,	0..8 {def.0}	SET_SUBNETB
"LM" ... Learning_Mode,	0..3 {def.0}	SET_LEARNMODE

0 .. no Learning, no Missing detection, legacy mode for LGRU16
1 .. no Learning, makes Missing detection for PIRs 1x received since power-up
2 .. needs Learning, ignores not-Learned PIRs, no Missing detection
3 .. needs Learning, ignores not-Learned PIRs, makes Missing detection

"tA" ... Time ALARM,	0..99 {def.2}	SET_TIMEA
"tt" ... Time TAMPER,	0..99 {def.2}	SET_TIMET

lengths of relay activity
value 0 ..... 250ms
values 1-60 ... 1-60s
values 61-99 .. (X-60) minutes (examples: 63 => 3 min, 99 => 39 min)

"r1" ... RELAY_1 Idle state,	0=N/O=Off, 1=N/C=On {def.0}	SET_IDLERE1
"r2" ... RELAY_2 Idle state,	0=N/O=Off, 1=N/C=On {def.0}	SET_IDLERE2
"r3" ... AUX_X1 Idle state,	0=N/O=Off, 1=N/C=On {def.0}	SET_IDLERE3
"r4" ... AUX_X2 Idle state,	0=N/O=Off, 1=N/C=On {def.0}	SET_IDLERE4
"LC" ... Light Control,	0..64 {def.0}	SET_LIGHTCTRL

0 ..... Light Control Disabled;
Detections work no matter if Day or Night
1-64 .. Light Control Enabled;
number is PIR Unit that gives the Day/Night information,
Detections work only on Night (also Masthead must be in SET state)

"tM" ... Test Mode, 0=Disabled, 1=Walktest_Mode_Enabled {def.0} SET_TESTMODE

===== Menu Level 2 [n2] ======= (Configuration of Tamper Relays) ======

"tP" ... Tamper,	0=Disabled,	1=Enabled {def.1}	SET_ENTAMPER
"CL" ... Cloak,	0=Disabled,	1=Enabled {def.1}	SET_ENCLOAK
"SH" ... Shock,	0=Disabled,	1=Enabled {def.1}	SET_ENSHOCK
"FA" ... Fault,	0=Disabled,	1=Enabled {def.1}	SET_ENFAULT
"MO" ... Move,	0=Disabled,	1=Enabled {def.1}	SET_ENMOVE
"MI" ... Missing,	0=Disabled,	1=Enabled {def.1}	SET_ENMISS
"bL" ... Battery_Low,	0=Disabled,	1=Enabled {def.1}	SET_ENBATTLOW
"cu" ... Code_Violation,	0=Disabled,	1=Enabled {def.0}	SET_ENCODEVIO
"Ld" ... Lid_Opened,	0=Disabled,	1=Enabled {def.0}	SET_ENLIDOPEN

===== Menu Level 3 [n3] ======= (Configuration of Global Relays==========

"F1" ... Function_1 (RELAY_1),	0..14 {def.1}	SET_FNC1
"S1" ... Source/Start Index for Function_1,	1..64 {def.1}	SET_SEL1
"E1" ... End Index for Function_1,	1..64 {def.64}	SET_END1
"F2" ... Function_2 (RELAY_2),	0..14 {def.2}	SET_FNC2
"S2" ... Source/Start Index for Function_2,	1..64 {def.1}	SET_SEL2
"E2" ... End Index for Function_2,	1..64 {def.64}	SET_END2
"F3" ... Function_3 (AUX_X1),	0..16 {def.9}	SET_FNC3
"S3" ... Source/Start Index for Function_3,	1..64 {def.1}	SET_SEL3
"E3" ... End Index for Function_3,	1..64 {def.64}	SET_END3
"F4" ... Function_4 (AUX_X2),	0..16 {def.15}	SET_FNC4
"S4" ... Source/Start Index for Function_4,	1..64 {def.1}	SET_SEL4
"E4" ... End Index for Function_4,	1..64 {def.64}	SET_END4
"AC" ... ARM/DISARM Control mode,	0..8 {def.7}	SET_ARMCTRL

0 .. no ARM/DISARM control (ALARM relays always work on Detections)
1 .. ARM/DISARM control enabled, via Keyfobs
2 .. ARM/DISARM control enabled, via Keypoints
3 .. ARM/DISARM control enabled, via Keyfobs & Keypoints
4 .. ARM/DISARM control enabled, via input INHIBIT (INHIBIT has priority)
5 .. ARM/DISARM control enabled, via input INHIBIT & Keyfobs
6 .. ARM/DISARM control enabled, via input INHIBIT & Keypoints
7 .. ARM/DISARM control enabled, via input INHIBIT & Keyfobs & Keypoints
8 .. ARM/DISARM control enabled, only via UART/USB command
Note: SET/UNSET commands sent via Keyfobs & Keypoints may optionally wait for confirmation (UART/USB command) from Alarm system.

"SC" ... Start Index for ARM/DISARM Control,	1..64 {def.1}	SET_STACTRL
"EC" ... End Index for ARM/DISARM Control,	1..64 {def.64}	SET_ENDCTRL
"rS" ... Restore Mode,	0..1 {def.1}	SET_RESTORE

0 .. restart with all relays Idle and Keyfob_state = SET (and KC_state Off)
1 .. restart with last known states (reload from EEPROM on power-up)

"EM" ... Extra Mode,	1..3 {def.1}	SET_EXMODE

1 .. print standard CALL & ALARM messages (Legacy mode)
2 .. print „Short texts“, (Control Unit mode)
3 .. print both, standard CALL & ALARM messages + „Short texts“ (debugging)

===== Menu Level 4 [n4] ===== (Commands for Learning Keyfobs & Reset) ======

"LE" ... Learn Keyfob	1 = do it	{def.0}
"Er" ... Erase ALL Learned Keyfobs	1 = do it	{def.0}
"dS" ... Default Settings	1 = do it	{def.0}

==========

Keyfobs

All Learned Keyfobs have the same function and are mutually interchangeable;
functions of individual buttons:
top-left ..... **SET (ARM)** // ALARM relays work on Detections
top-right ..... **UNSET (DISARM)** // ALARM relays do NOT work on Detections
bottom-left ..... **KC_ON** // KC_ON/OFF can be mapped to one output
bottom-right ..... **KC_OFF** // (and be used as an independent switch)
Note: two bottom buttons were spare, I decided to use them for control of signal KC (Keyfob_Controlled), any of the 4 available outputs (2 relays + 2 open-drain) can be configured to output this signal.
Can be used for example for switching some light ON/OFF, whatever..
Or can be left unused.

How to Learn a new Keyfob:

User must switch to „Learning of Keyfobs“,
- either enter command **LEARNKF** ,
or do it via Menu Level 4 / "LE" (Learn Keyfob), change value to 1 end hold MEM button for >3s.
Now press any button on the Keyfob, it will be Learned.
Stop „Learning of Keyfobs“ via command **STOPLEARNKF** , or make power cycle.

```
learnkf                 // entered command LEARNKF
ACK
Learning Keyfobs..
<KF_279C78B6_0>         // printed after transmission from a Keyfob, any button
new [00]                // shows on which position of memory was stored
stoplearnkf             // entered command STOPLEARNKF
ACK
Learning Keyfobs done   // now is in normal operation mode

<KF_279C78B6_0>         // pressed top-left button (to SET, but was already SET)
[KF_S_S]                // Keyfob, wanted_SET, made_SET
KF answer cancelled

<KF_279C78B6_2>         // pressed top-right button (to UNSET)
[KF_U_U]                // Keyfob, wanted_UNSET, made_UNSET
[UNSET]                 // state has changed, shows new state
KF answer cancelled

<KF_279C78B6_1>         // pressed bottom-left button
[KC_ON]                 // it is an independent switch, turned On
KF answer cancelled

<KF_279C78B6_3>         // pressed bottom-right button
[KC_OFF]                // it is an independent switch, turned Off
KF answer cancelled
```

Table of available Functions

Value Function

---

00 ... MOS-relays: no_function, relay permanently Off (=> stays Off on power-failure)
Open-Drain-Outputs: no_function, high impedance (NFET transistor is Off)
01 ... Group_Alarm, group is selected via Sx..Ex
02 ... Group_Tamper, group is selected via Sx..Ex
03 ... Battery_Low, group is selected via Sx..Ex
04 ... Battery_Low, group is selected via Sx..Ex + IR_tamper
05 ... Missing, group is selected via Sx..Ex
06 ... Missing, group is selected via Sx..Ex + IR_tamper
07 ... Light_Control, source is selected via Sx (independent on Menu param "LC")
08 ... Light_Control, source is Menu parameter "LC"
09 ... SET_output, source of SET/UNSET state depends on Menu param "AC" ARMCTRL
10 ... reserved
11 ... reserved
12 ... KCON_output, controlled via two bottom buttons of Learned Keyfobs
13 ... IR_tamper, active = Lid Opened
14 ... Keypoint_event, Sx=1 BAD_CODE // Keypoint used 3x with wrong PIN
Sx=2 KEYPOINT_TAMPER // position change by accelerometer
Sx=3 BAD_CODE + KEYPOINT_TAMPER
Sx=4 IR_tamper + BAD_CODE
Sx=5 IR_tamper + KEYPOINT_TAMPER
Sx=6 IR_tamper + BAD_CODE + KEYPOINT_TAMPER
Sx=others like 6

---

for F3 and F4 only:
15 ... input INHIBIT, with Pull-Up, // connect to GND to inhibit ALARMS
16 ... input INHIBIT, with Pull-Down, // connect to >= +3.3V to inhibit ALARMS

- default function F1 assigned to **RELAY_1** is 01 ... Group_Alarm (group PIRs 1..64 => Global) F1=1, S1=1, E1=64
- default function F2 assigned to **RELAY_2** is 02 ... Group_Tamper (group PIRs 1..64 => Global) F2=2, S2=1, E2=64
- default function F3 assigned to **AUX_X1** is 09 ... SET_output
F3=9, S3=1, E3=64
- default function F4 assigned to **AUX_X2** is 15 ... input INHIBIT, with Pull-Up
F4=15, S4=1, E4=64

Note, for Open-Drain-Outputs:
- options 01..14 are Open-Drain outputs,
Idle state is when NFET transistor is Off (no Drain current, Drain floating),
Active state is when NFET transistor is On (Drain switched to GND, current up to 3A, 20V max)

Mapping of functions to outputs
MOS-relays and Open-Drain outputs

Menu Level 3 contains four groups of parameters:

F1/S1/E1 …….. F1 selects function for MOS Relay 1 (terminals RELAY_1),
S1 to E1 is group of units included / or S1 is selected source
F2/S2/E2 …….. F2 selects function for MOS Relay 2 (terminals RELAY_2),
S2 to E2 is group of units included / or S2 is selected source
F3/S3/E3 …….. F3 selects function for terminal AUX_X1 (can be Open-Drain output or Input),
S3 to E3 is group of units included / or S3 is selected source
F4/S4/E4 …….. F4 selects function for terminal AUX_X2 (can be Open-Drain output or Input),
S4 to E4 is group of units included / or S4 is selected source
Functions that can be mapped to RELAY_1 / RELAY_2 / AUX_X1 / AUX_X2 are listed on next page.
Table contains 17 options (functions).
The first 15 options are common for all 4 terminals.
But there are 2 more options available only for terminals AUX_X1 or AUX_X2, because these can serve not only as Open-Drain Outputs, but also as Inputs.
MOS-relays are galvanically isolated outputs, can not serve as Inputs.

Indicators

MOS-relays / Open-Drain-Outputs and LED indicators are driven separately.
Meaning of all LED indicators is always ACTIVE (LED is On) / IDLE (LED is Off), no matter whether the particular MOS-relay or Open-Drain-Output is configured NO or NC.
Only states of MOS-relay and Open-Drain-Outputs is affected by NO/NC setting.
LED-indicator SIGNAL is On for 250ms on reception of valid Genesis packet, no matter if signal comes from PIR of any Site/Subnet/Unit or Keyfob or Key-point.
Useful for verification the PIR or Keyfob or Keypoint you test really works.
LED-indicator POWER is permanently On to indicate Masthead is operating.