

Appendix A

Timer Firmware Description

The aim of this appendix is to give a detailed explanation of the logic implemented within the timer unit firmware. The diagrams that follow closely mirror the source code, but skip some of the unnecessary details involved in each block step.

The logical place to start is at the beginning, and so Figure A.1 begins with the unit receiving power, and details the unit initialization and the main program loop.

The timer uses the DTR serial control line trigger a hardware reset remotely from the Acquisition PC. This simplifies many details, but has one notable downside - the unit is forced to reset whenever software on the Acquisition PC first opens a connection via USB¹.

If data is received over the USB connection within the first 4 s after power-on, the unit will enter its update mode, allowing diagnostics and firmware updates using the AVR-109 protocol at 9600 baud. The update mode can be exited by sending the ascii character 'E' or toggling the DTR line to trigger a hardware reset.

A work-around is required to support relay mode, as the PC GPS software does not wait the required 500 ms before sending data, and would otherwise force the unit into the firmware update mode. This is achieved by using the `timerutil` program (see Appendix ??), to send a command packet which enables a one-time bypasses of the boot-loader, causing the unit to boot directly into relay mode. The bypass is only valid during the next hardware reset, ensuring that the unit cannot become stuck in relay mode.

The main program begins after the boot-loader timeout², which initializes the hardware components and sends configuration messages for the Trimble and Magellan

¹This is strictly true only for Linux and OS X, but we force additional resets to ensure consistent behavior on all platforms.

²To avoid ambiguity, i'll use timeout to refer to actions run via the microprocessor timer/counter subsystem.

GPS receivers via the serial port.

After initialization, the main program loop runs indefinitely:

- Message flags set by interrupts and USB/GPS subroutines are read and (unless relay mode is active) the appropriate data is sent to the Acquisition PC.
- The camera logic input is polled, and a $500\ \mu\text{s}$ timeout is set if a change is detected to ensure the level change isn't caused by a spurious noise pulse. Any actions that run as a consequence of this level change occur within the logic monitor timeout interrupt.
- USB commands are processed (Figure A.2).
- GPS messages are processed (Figure A.3).

Hardware or software interrupts may trigger any time during the main loop, and will take over the processor until they complete. Interrupt sources include:

- GPS and USB serial reads/writes occur asynchronously, using interrupts to copy bytes into a buffer as soon as they are received, and a second buffer which data is copied to for transmission in parallel with the main loop.
- A timeout used to restore the output trigger level $512\ \mu\text{s}$ after it begins.
- A timeout used to set the GPS status to `UNAVAILABLE` after 3 s without data.
- A timeout used to verify and act on camera logic output change (Figure A.5).
- A timeout used to increment the high-speed exposure counter (Figure A.6 and explained below).
- A hardware interrupt on the falling edge of each 1 Hz input pulse (Figure A.4 and explained below).

Two timing modes are supported, with different resolution and maximum exposure times. The pulse-counting mode provides a 1 s resolution between 1 - 65535 s (18.2 hours). A counter is decremented on each 1 Hz time pulse and, when it reaches zero, a trigger pulse is sent to the camera and a flag (`record_trigger`) is set to allow the next serial timestamp to be sent as the trigger time to the Acquisition PC.

It is assumed that the first timestamp received after a trigger pulse is sent corresponds to the trigger. An incorrect trigger timestamp may be sent in the (rare) event that the serial data stream is corrupted, and the correct timestamp lost. The output trigger pulses are delayed by $19\ \mu\text{s}$ from the input pulse. This offset is consistent between exposures, and so it is assumed to be negligible given the 1 s exposure resolution.

The high-speed timing mode provides a 1 ms resolution for exposures between 1 - 65535 ms (11 minutes), but requires the GPS receiver to provide a stable 10 MHz output signal from which to run the processor (this is supported by the Trimble Thunderbolt and several other GPS receivers, but not the Magellan receivers at VUW or MJUO).

A switch on the back of the unit selects between the internal oscillator and the external clock signal, which is then divided using the timer/counter subsystem within the processor to produce an interrupt accurately aligned to each millisecond.

This millisecond-interrupt decrements the exposure counter and updates an internal estimate of the current millisecond time component. The millisecond counter is allowed to increment past 1000 ms, which adds a significant degree of robustness against glitches in the GPS serial stream - the timer will internally correct up to 65 continuous missing serial timestamps before timing accuracy is lost. The counter is reset modulo 1000 each time a serial time packet is received.

In high-speed mode, the 1 Hz signal is used as a check on the internal timing, by verifying that the millisecond counter is an integer multiple of 1000.

The timer is started with an initial count value chosen to reduce the time offset between the 1 Hz input pulses and the output trigger to within 500 ns. A future improvement may make this delay customizable in order to correct an exposure phase shift introduced by the triggered operation of the ProEM camera.

Three input command packets are recognized by the timer. The `ENABLE_RELAY` command is sent by the `timerutil` utility to place the unit into relay mode. In relay mode, all data sent via USB is forwarded to the GPS receiver, and vice-versa, and so the timer acts as a USB-to-serial adaptor and allows GPS configuration software to be run on the Acquisition PC. The 1 Hz input pulses are also forwarded to the camera, but note that the polarity and pulse width are different to the outputs of the Trimble and Magellan receivers.

The `START_EXPOSURE` command is used to start an exposure sequence. The command packet includes the timing mode to use, whether to monitor the camera logic output or simulate the logic changes internally, and the exposure time. The unit will wait until the camera is ready to start exposing (signalled by the logic output going low), and will then wait a further period of time to ensure that exposure transitions will occur on the round minute (assuming an exposure period that evenly divides into 60 seconds). If pulse-counting mode is active, the first exposure will then begin immediately, but the high-speed mode requires an additional exposure period to ensure the correct exposure length. The first exposure (which is discarded by the acquisition software) may therefore be up to twice as long as the requested exposure

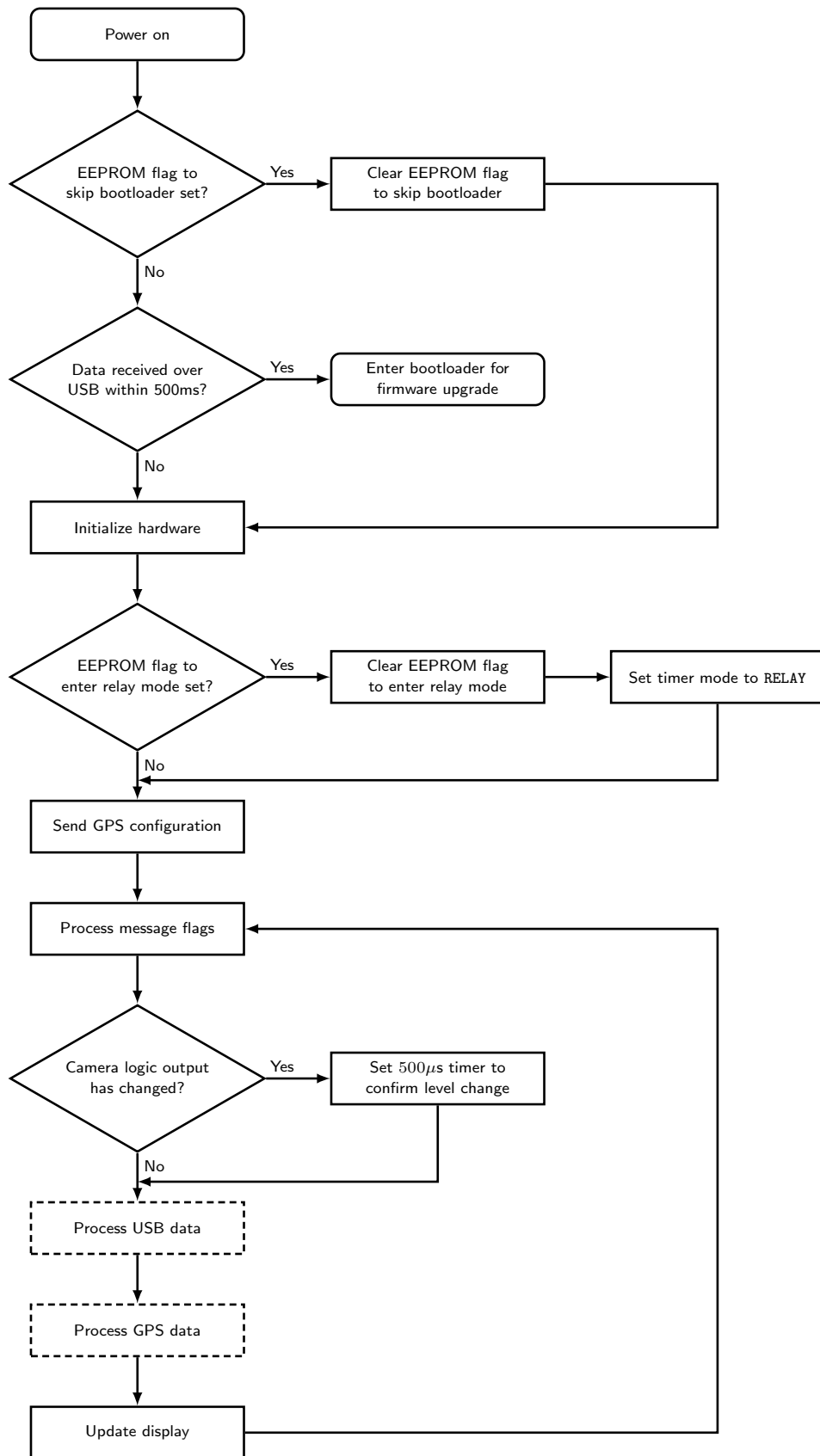


Figure A.1: The main loop handles initializes the hardware and handles non-time-critical functions. Each of the dotted boxes are expanded in additional figures.

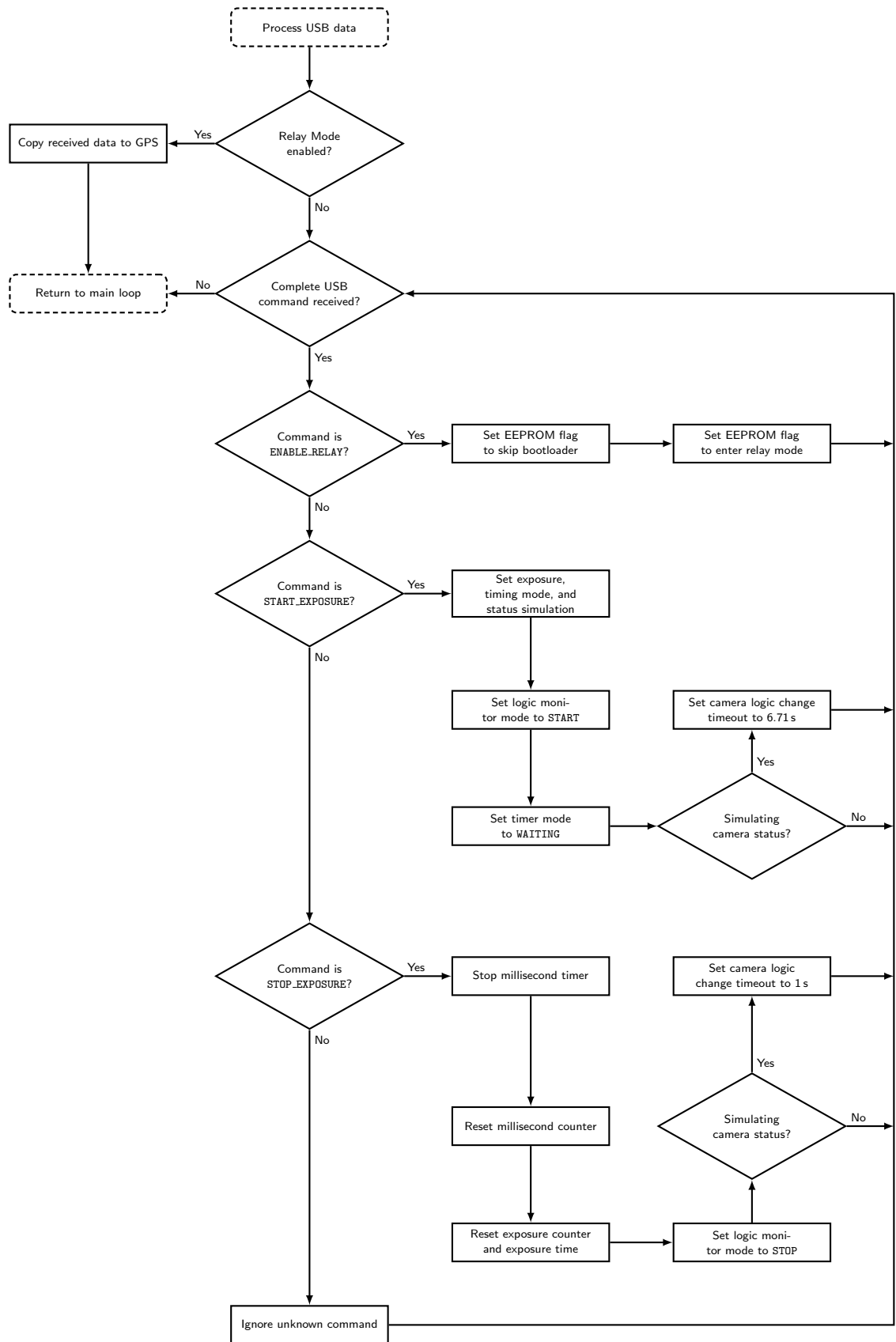


Figure A.2: Expanded detail of the USB command processing logic.

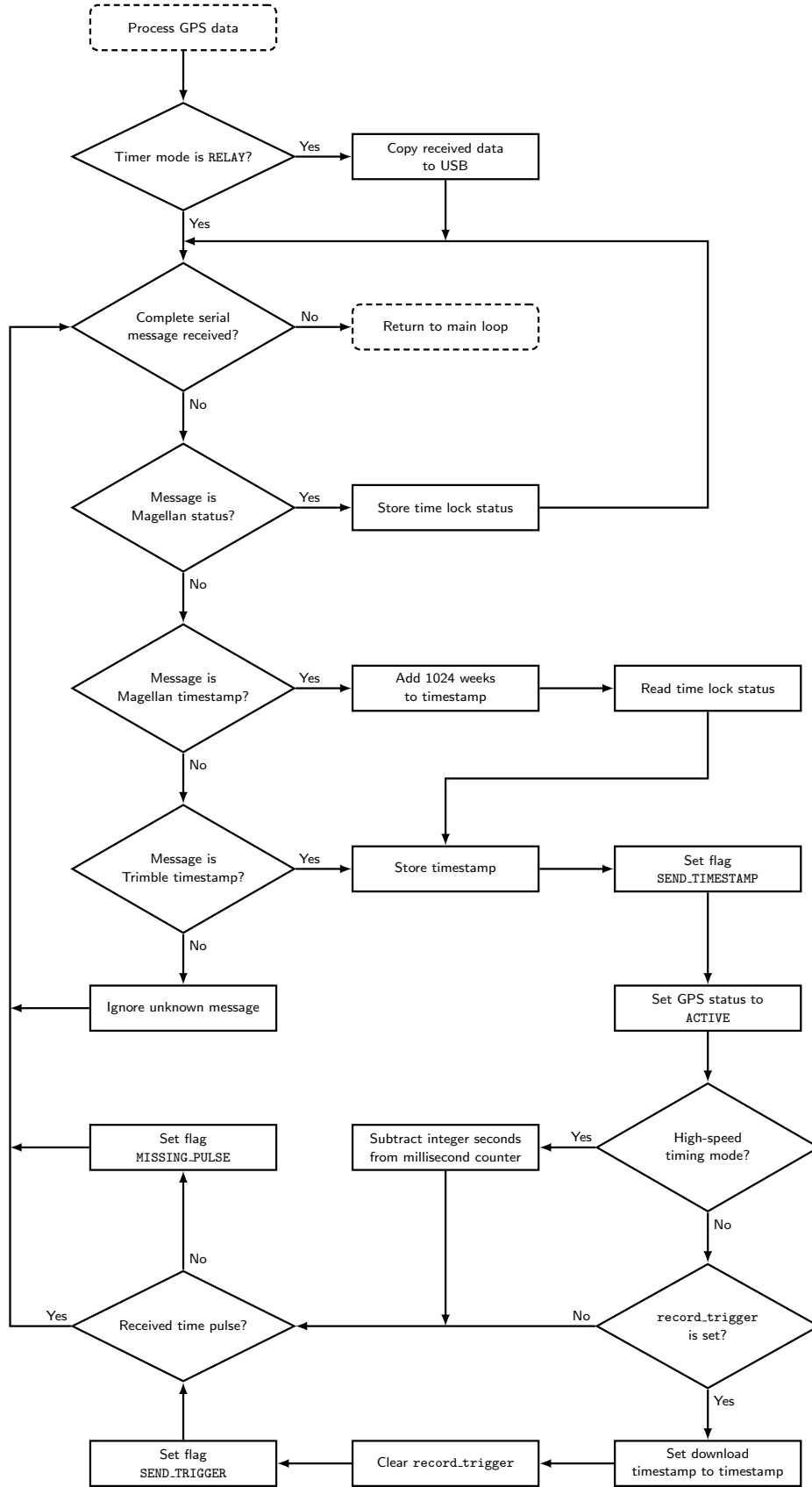


Figure A.3: Serial messages from the GPS are processed to extract the current time and satellite lock status. A check is in place to ensure that serial timestamps and time pulses are received in alternating order.

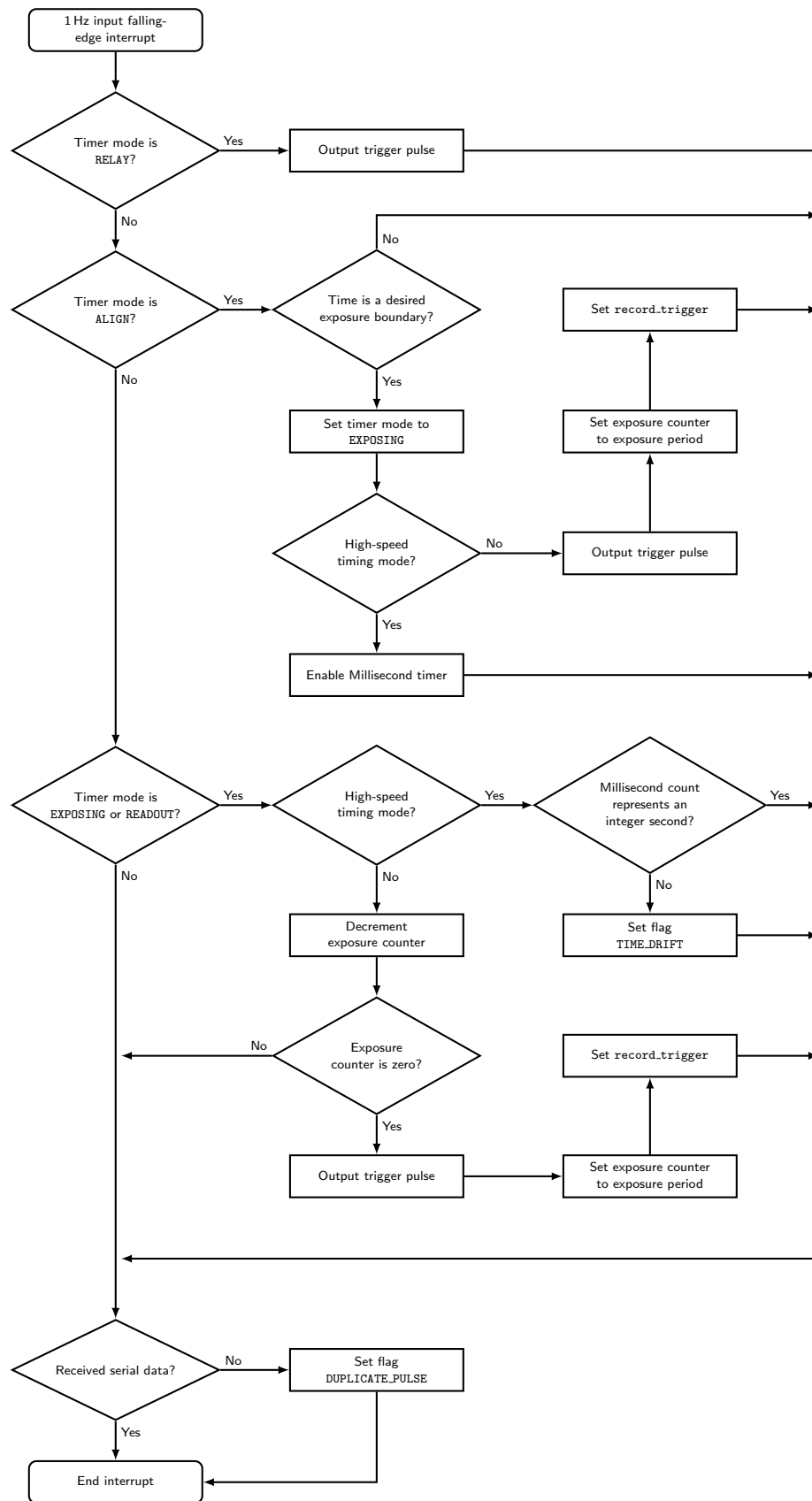


Figure A.4: A pulse on the 1 Hz input line triggers different actions depending on the current timer mode. See text for details.

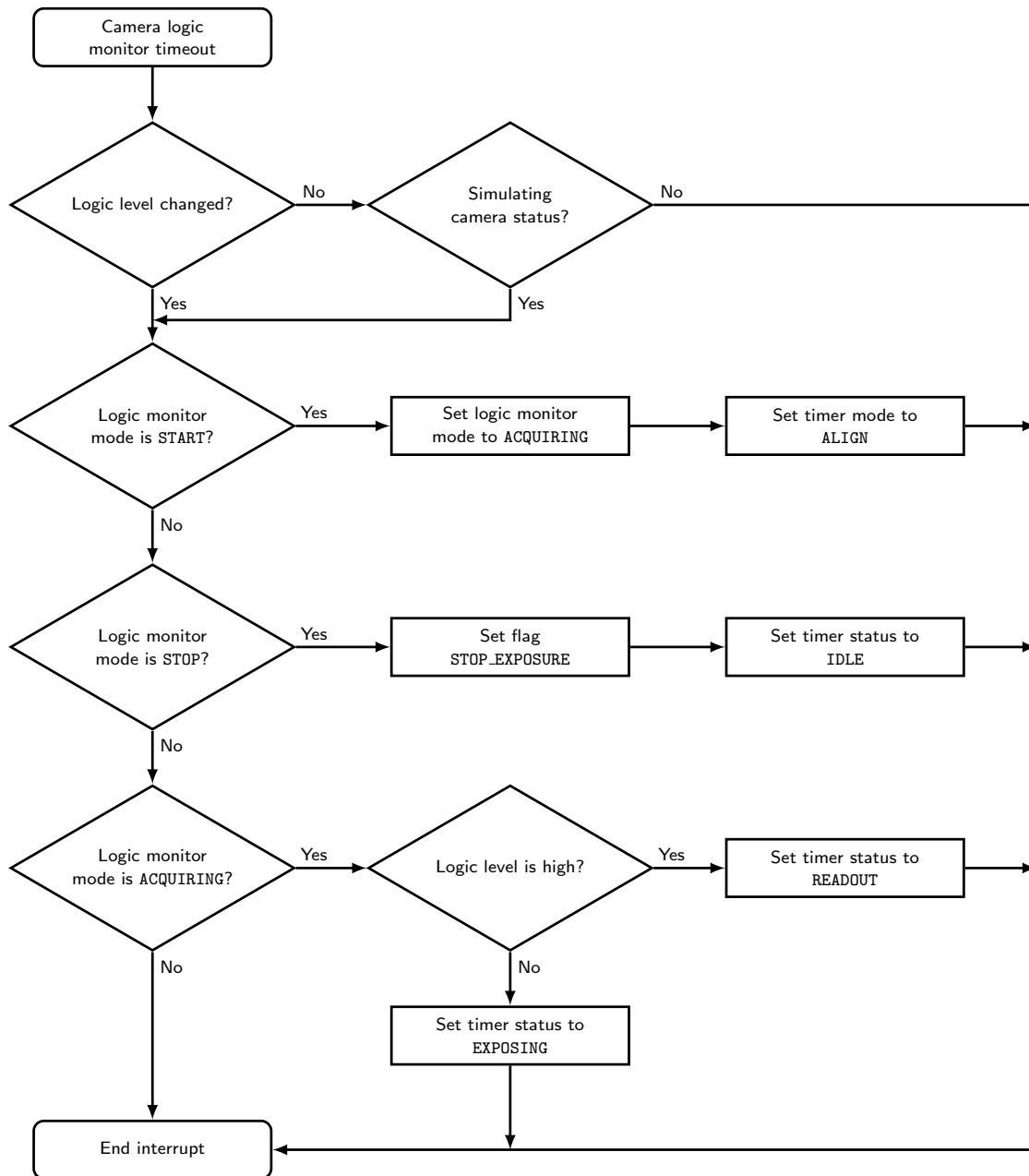


Figure A.5: A block diagram showing the camera monitoring logic.

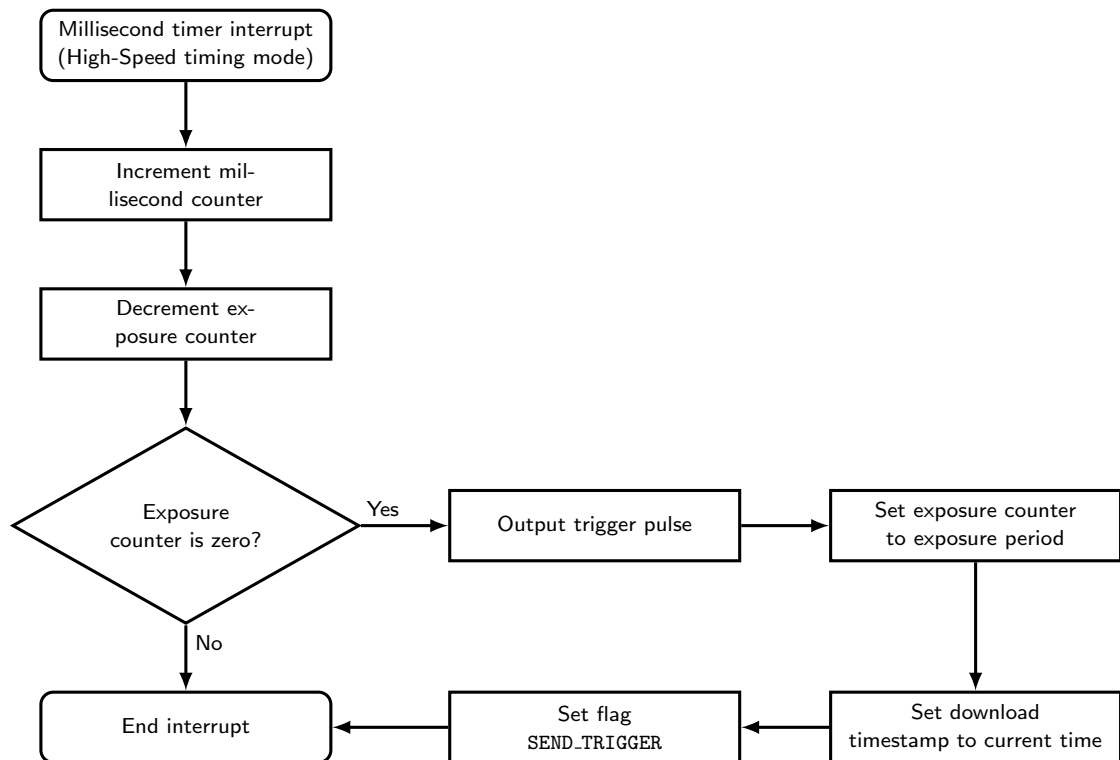


Figure A.6: The millisecond timer is used in the High-speed timing mode as a high-cadence equivalent to the 1 Hz input.

time, which may cause bright objects in the field to over-expose.

The `STOP_EXPOSURE` command immediately stops the exposure countdown and sets the logic level monitor to notify the Acquisition PC when it is safe for the Acquisition PC to stop the exposure sequence (a serious memory-corruption bug in the PVCAM camera software library is triggered if an exposure sequence is stopped while a camera readout is progress).