output matrix. As soon as the LED is turned on, the controller U7 begins a high-speed sampling of the results by examining the output pulse from the logically associated IrDA receiver IR1-IR12.

[0052] Thirty-two samples are taken at ⅛th microsecond intervals and a profile of the IrDA receiver output (if any) is compiled and stored in memory. This scanning occurs at the highest priority processor interrupt level (internally timer-driven) so that it is guaranteed not to be interrupted. The "output profile" is roughly analogous to an oscilloscope trace. It is a time-record of the output waveform from the IrDA receiver. As described further below, a touchframe firmware optical noise immunity algorithm evaluates the profile. The algorithm looks to see if the pulse present in the output profile began too soon—indicating that some optical noise other than the LED beam triggered the IrDA receiver—or else if the pulse width is too short or non-existent, indicating that the beam is blocked.

[0053] The beam may appear to be blocked for several possible reasons, i.e., it may be blocked by a finger or stylus, which is normal operation, or it may be blocked by debris or some other obstacle, or it may indicate a burned out LED **24** or burned out IrDA receiver **42**. Other "layers" of the firmware evaluate the difference among these possibilities and take appropriate action.

[0054] The memory profile of the IrDA receiver **42** is very transient—it is only used to evaluate the quality of the firing of a single beam from a selected LED **24**. Once the evaluation is complete, the resulting information is stored in a more compact form in a state machine associated with the particular logical beam. The profile itself is overwritten immediately, and a short time later the beam of the next selected LED **24** is profiled and evaluated.

[0055] LED Drivers

[0056] With continued reference to **FIG. 8**, dual field effect transistors (FETs) U19-U49 are arranged in a push-pull arrangement to avoid having any rows or columns of the matrix float. This provides clean switching transitions, preventing spurious LED outputs. As previously mentioned, only one LED **24** is illuminated at a time.

[0057] Optical Noise Immunity

[0058] The touchframe firmware optical noise immunity performs a "sanity check" on each input received by the controller **44** from an IrDA receiver **42** by comparing the actual input profile, i.e., the output profile of the IrDA receiver, with an expected input profile. There is a three-state logic applied. Each beam is treated as being in one of three instantaneous states: Connected, Blocked, or Noise. A beam evaluated as Noise does not affect the state counter. A beam evaluated as Connected or Blocked moves the state counter in the direction of more connected or more blocked, up to a predetermined limit. This provides some hysteresis in transitions between Connected and Blocked states as seen by the next level of system firmware. Furthermore, as described in detail later, a state counter is used to "debounce" the beam state.

[0059] With reference to **FIG. 9**, wherein the scale from left to right represents a time period of four microseconds subdivided into ⅛th microsecond intervals, each ⅛th microsecond the controller **44** fires one of the LEDs **24** in the array and then immediately begins to sample the output from the associated IrDA receiver **42**. There is an aggregate delay that includes propagation delay from the controller **44** through the MUXing logic, time required to switch on the LED **24**, time the IrDA receiver **42** takes to respond to the increase in light and time required for the output to propagate back to the controller. Finally the signal is logged internally by the controller **44**. If, however, a signal is detected before an adequate amount of time has passed, the noise rejection algorithm assumes that some external source of light other than the LED **24** that was fired caused the IrDA receiver **42** to trigger. Waveforms of five example beams considered Noise, Good, or Blocked are illustrated.

[0060] A beam is rejected as Noise if there is an output pulse edge present in the profile at a time earlier than is expected, i.e. earlier than the IrDA receiver **42** response time would allow. This time is noted as the "noise threshold." A beam is accepted as Connected only if the pulse edge occurs after the noise threshold and pulse width is a reasonable length All other beams are considered Blocked.

[0061] It is highly unlikely that in any normal environment there would be infrared noise of sufficient intensity and duration that would interfere with the operation of the touchframe system. Infrared noise of such magnitude would block the operation of any conceivable infrared touchframe technology. The touchframe optical noise immunity feature of the invention has been tested by observing operational output in the presence of infrared noise generated by various remote control devices.

[0062] Minimizing Optical Noise Emission

[0063] As previously mention, prior art touchframe systems may interfere with external infrared receivers, such as those in a TV receiver. In order to avoid the possibility of the touchframe system interfering with such external infrared receivers, the controller **44** is programmed to activate LEDs **24** at pseudo random intervals and in a pseudo random sequence. This causes any infrared pulses that may stray into the environment to appear to be "noise" to any other infrared actuated device.

[0064] Error Handling and Redundancy

[0065] Each touchframe system begins life fully operational as guaranteed by in-circuit test. The in-circuit test fixture may also load code into the touchframe system controller **44**, which is FLASH-based. No adjustments or calibration are needed. The touchframe system is designed to accommodate the full range of possible component tolerances and ambient lighting conditions.

[0066] The touchframe system constantly tests itself by firing and detecting infrared beams. When a beam is blocked for an extended period of time the software state counter toggles into a Long Term Blocked state and that beam receives special software handling. This extended period of time maybe set in the touchframe firmware and may be, for example, approximately 30 seconds.

[0067] As previously described with reference to **FIG. 4**, the touchframe has six rows of LEDs. Each row faces a pair of IrDA receivers **42** at the opposite side of the touchframe. The presence of pairs of IrDA receivers **42** provides the touchframe system a level of redundancy. Firmware algo-