IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 10, October 2026

DOI: 10.17148/IJIREEICE.2026.141024

A surface tilt monitoring and controlling system, with additional time-based control and alarm indicators, utilizing mercury tilt switches, electromagnets, and FPGAs, programmed with VHDL and suitable for industrial and laboratory applications

Dr Evangelos I. Dimitriadis[1], Leonidas Dimitriadis[2]

Department of Computer, Informatics and Telecommunications Engineering, International Hellenic University, End of Magnisias Str, 62124 Serres Greece[1]

Department of Information and Electronic Engineering, International Hellenic University, 57400, Sindos Thessaloniki, Greece[2]

Abstract: A novel surface tilt monitoring and controlling system, which uses FPGAs and VHDL, is presented here. The system monitors continuously the left or right tilting mode of a surface equipped with metal strips. In case that tilting occurs one of two mercury tilt switches also existing on the surface, produces output logic 1, leading to corresponding electromagnet activation, in order to restore previous horizontal balance of the surface. Right or left tilt switch is activated after respective right or left tilting of the surface. Simultaneously right or left half of FPGA's board LEDs and seven-segment displays are activated, in order to present the tilting direction, while electromagnet's LED activation provides additional proof for its appropriate operation. Another process being activated upon tilting, is the one that measures time period of surface tilting mode. If time period set by programmer is exceeded, red LED alarm is activated indicating malfunction of electromagnets or other part of the system, proving it unable to restore previous horizontal balance of the surface. Our system is also equipped with a process that prevents both electromagnets and time measuring operations to proceed, in case that tilting occurs. This can be achieved by turning FPGA board SW0 switch to logic 1 state and gives the advantage of unloading surface mode, after intentionally causing the surface to tilt. Our system is cheap to manufacture, easy to use and due to FPGA incorporation can be combined with IoT and AI systems, in order to achieve monitoring and controlling of a large number of industrial product elevator systems, or platform surfaces for laboratory applications.

Keywords: surface tilt monitoring and controlling, mercury tilt switches, electromagnets, FPGA, VHDL, indicating LEDs, time period of tilting, alarm LED.

INTRODUCTION

It is well known that FPGAs have attracted attention of a great number of researchers in the recent years, for industrial as well as for a variety of other applications. [1-15] FPGAs have the main advantage of combining software and hardware, thus enabling hardware programming for a series of applications. The most used languages for FPGAs' programing are VHDL and Verilog and VHDL is the one used in our work.

Although a lot of work has been done concerning implementation of FPGAs in a variety of applications as mentioned above, we found only a few works[16-18] dealing with tilt systems incorporating FPGAs. They concern Tip-Tilt Mirror Control Based on FPGA for an Adaptive Optics System, as well as Automatically controlled pan–tilt smart camera with FPGA based image analysis system dedicated to real-time tracking of a moving object, while the third one presents Design of A High-Precision Component-Type Vertical Pendulum Tiltmeter Based on FPGA.

There are no works concerning surface tilt systems by using mercury tilt switches, electromagnets and FPGAs. An FPGA-based system uses the high-speed processing and precise timing capabilities of FPGAs and it is highly useful in scientific, industrial, and consumer applications where real-time analysis and high-throughput signal digitization are

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 10, October 2026

DOI: 10.17148/IJIREEICE.2026.141024

required. Monitoring and controlling surface tilt using an FPGA (Field Programmable Gate Array) is an excellent choice for applications that require high speed, low latency, and parallel data processing. It is used in camera stabilization systems (gimbals), aerospace, robotics, and solar panel alignment.

Additionally mercury tilt switches offer several unique physical and electrical advantages when used in tilt-sensing systems:

• Low electrical resistance: Even a small amount of mercury ensures excellent conductivity and minimal contact resistance.

• High current-carrying capacity: They can handle and carry significant amounts of electrical current despite their small size—unlike many modern low-power sensors.

• Self-cleaning contacts & long service life: Because the contacts are hermetically sealed (often in an inert gas or a vacuum), they do not oxidize. Even if a small internal arc occurs, the mercury surfaces are effectively renewed with each operation.

• High surface tension: The high surface tension of mercury creates a cohesive droplet, ensuring a stable and clean contact with the electrodes.

• Silent and smooth operation: The movement of the mercury involves no metal components striking one another, rendering the operation completely silent.

• Prevention of sparks in hazardous environments: The circuit interruption occurs within a sealed glass enclosure, preventing the emission of sparks that could ignite flammable gases or dust (a historical advantage in industrial or hazardous zones).

• Simplicity and no calibration required: They do not require complex electronics, software, calibration, or amplifiers, offering straightforward, gravity-based on/off operation.

• Visual visibility: The silvery, lustrous appearance of mercury allows for easy visual inspection of the switch's state when housed in transparent enclosures.

We present here a novel system able of monitoring and controlling tilting for a variety of surfaces, by using mercury tilt switches, electromagnets, FPGAs and time based alarm system. Our system is equipped with an unloading surface process and can also provide information to a central AI based system or send useful data via 5G system for remotely monitoring and controlling a large number of surfaces used in industrial or laboratory applications.

DESIGN OVERVIEW AND OPERATION OF THE SYSTEM

Figure1 presents device overview and operational units of our system, using FPGA DE10-Lite board, while Figure 2 presents circuit diagram of the system. Subsequent Figure 3a, b presents the implemented surface tilt monitoring and controlling system of this work.

It is obvious from the above figures that our system, except from DE10-Lite FPGA board, contains also some basic circuit parts. First of all we use a breadboard playing the role of the tilted surface. It also uses two KY017 mercury tilt sensor switches which are placed upon the breadboard. They are the main tilt monitoring units providing FPGA board with a logic 1 signal input whenever tilting to the left or right occurs. This signal activates corresponding electromagnet in order to exert attractive force to metal strips, also screwed near breadboard's edges, for achieving former horizontal direction of the breadboard. The system uses two electromagnets acting as outputs and placed on two bases above breadboard, one to the right and the other to the left edge of the breadboard. Both mercury tilt switches and their corresponding electromagnets need 5V voltage supply for operation, provided by the FPGA board. In case that no tilting occurs, both red lights on mercury tilt switches modules are ON, while in the opposite case tilted switch's light is OFF.
We also use two external white LEDs one for each electromagnet and their role is to light up upon activation of corresponding electromagnet. Finally an external time alarm red LED is used, for indicating that after 3 seconds of tilting left or right, no previous horizontal balance of the breadboard is achieved leading to the conclusion that malfunction problem to electromagnets has occurred. Needless to mention that all LEDs used in our system are protected with a 330 Ohm series resistance and they are connected to I/O pins of the FPGA board acting as outputs for our system. FPGA's board LEDs and SW0 switch are also used in our system.

Time is the other input value used here and it is provided by FPGA's clock. It can also be seen in Figure 3a presenting our handmade tilting system, that breadboard with all its components is hung with thin ropes from the bases of the electromagnets, providing it with the ability of tilting.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 10, October 2026

DOI: 10.17148/IJIREEICE.2026.141024

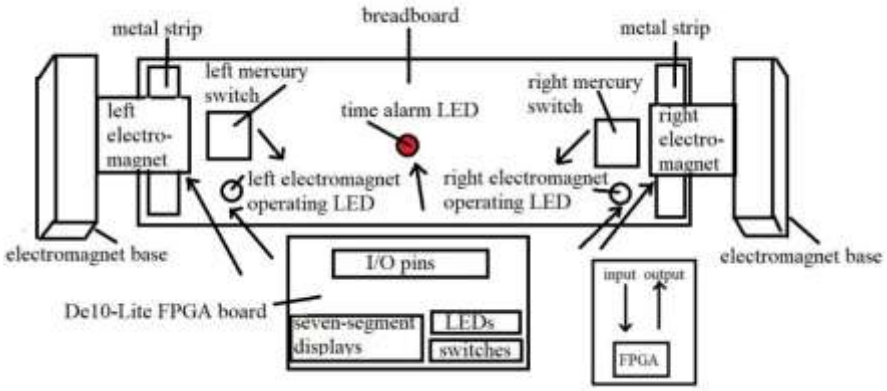


Figure 1: Device overview and operational units of our system.

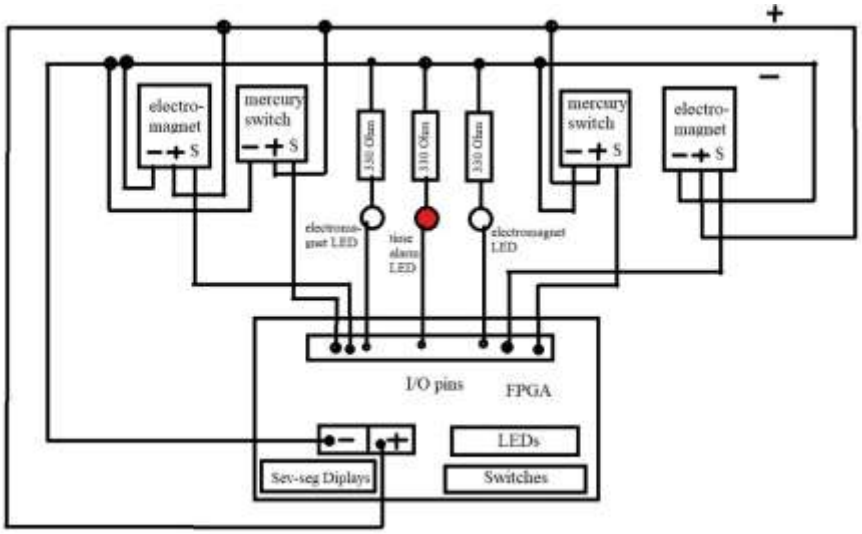
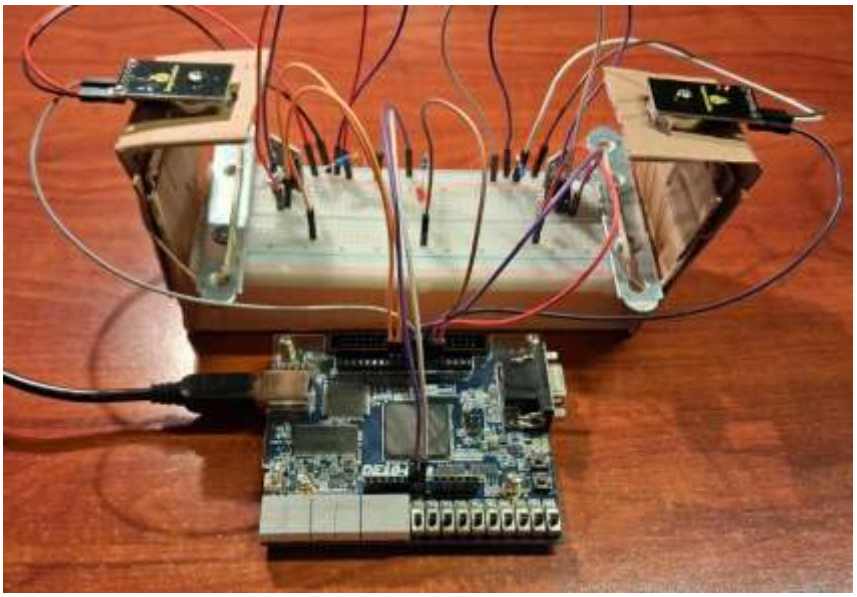


Figure 2: Circuit diagram of our system



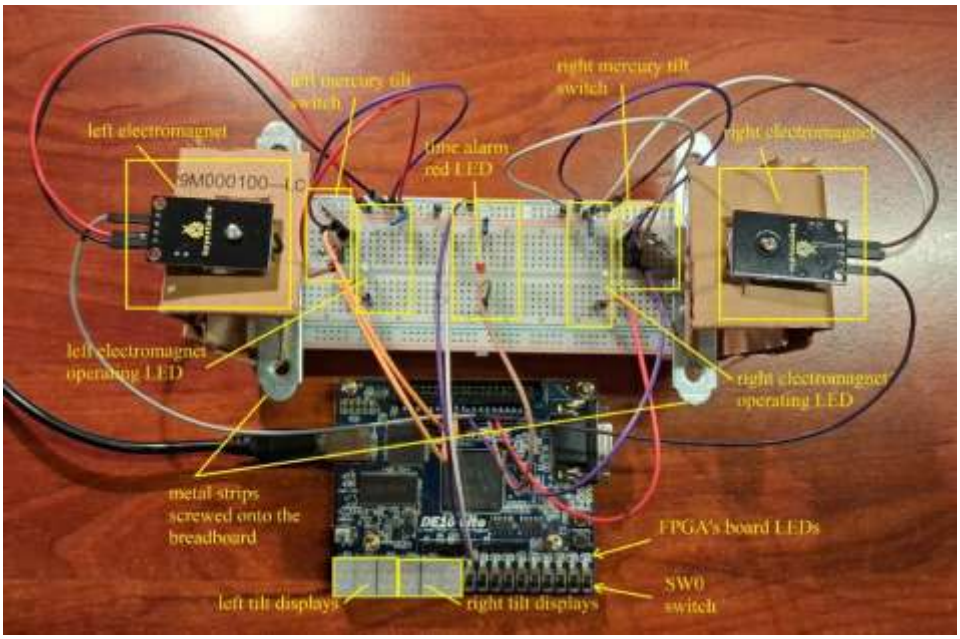
a)

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 10, October 2026

DOI: 10.17148/IJIREEICE.2026.141024



b)

Figure 3a, b: The implemented surface tilt monitoring and controlling system of this work a) without notifications and b) with explaining notifications.

Our surface tilt monitoring and controlling system starts operating as soon as power supply +5V is applied to the circuits via DE10-Lite appropriate pins and the VHDL program is sent via USB Blaster interface, to FPGA chip.
Input values from both KY017 mercury tilt sensor switches enter our system. If the breadboard surface is horizontal then no tilt occurs and both tilt switches send an output logic value 0 as input to FPGA board. This situation is presented in Figure 4 where the surface tilt monitoring and controlling system is in operation mode but no tilt is detected in both switches (their red light is ON), thus no electromagnet is needed to be activated and seven-segment displays present this balance mode by displaying "– – – – – – ". It is obvious that the above horizontal balance mode is the final and also immediate result of corresponding electromagnet activation, in case that tilting to the left or right direction occurs.

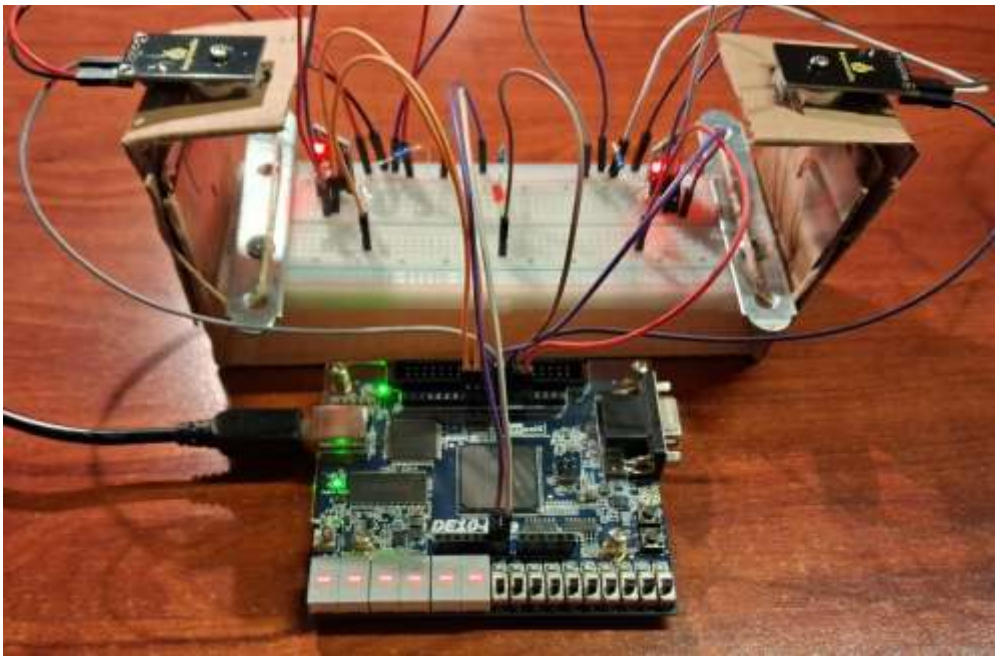


Figure 4: The surface tilt monitoring and controlling system in operation mode. No tilt is detected in both switches (their red light is ON), thus no electromagnet is activated and seven-segment displays present this balance mode.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

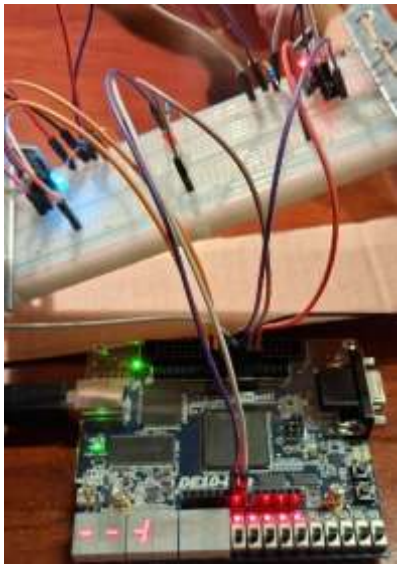
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 10, October 2026

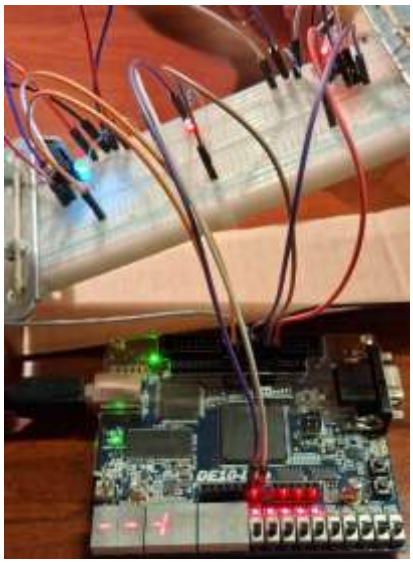
DOI: 10.17148/IJIREEICE.2026.141024

The tilting conditions are presented in Figures 5 and 6.

Figure 5 shows the surface tilt monitoring and controlling system in operation mode. In 5a the breadboard surface is tilted to the left, leading to immediate left mercury switch activation (red light of the module is OFF) and consequently to left electromagnet activation (its white LED is ON) by sending via FPGA I/O pins logic output 1 to left electromagnet. Simultaneously half left of board LEDs and three left seven-segment displays are ON, providing an indication of the above tilt. In 5b breadboard surface is mechanically forced to remain tilted with consequent red LED alarm activation after 3 seconds. SW0 board switch is OFF and combined with tilt mode of the breadboard, these conditions activate time measuring process by using FPGA's clock and allow us to measure time interval between starting the tilting mode and reaching the critical time value of 3 seconds remaining tilted. If the above time value is exceeded simultaneously red LED alarm lights up indicating an operating problem probably to corresponding electromagnet. Depending on the application of our system, programmer could set the above time period of remaining tilt, as appropriate.



a)



b)

Figure 5: The surface tilt monitoring and controlling system in operation mode: a) breadboard surface is tilt to the left, thus left mercury switch is activated leading to left electromagnet activation (its white LED is ON). Simultaneously half left of board LEDs and three left seven-segment displays present the above tilt. b) breadboard surface is mechanically forced to remain tilted with consequent red LED alarm activation after 3 seconds.

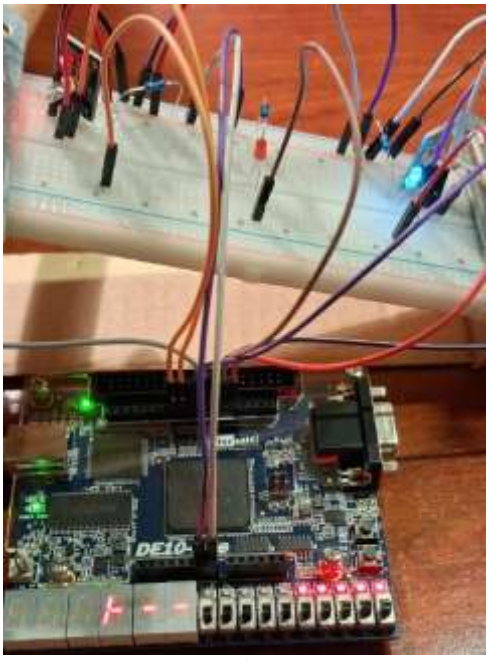
IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

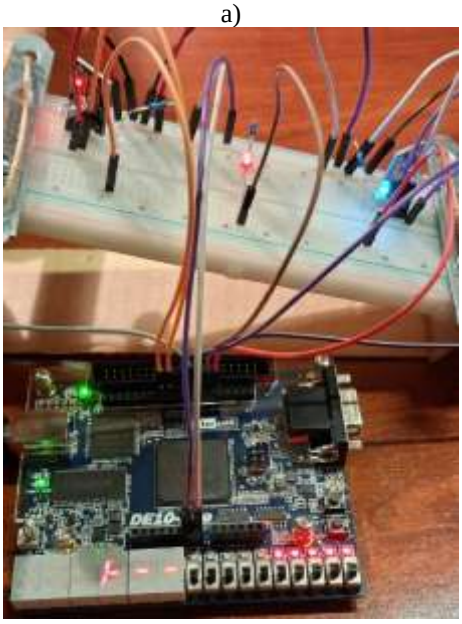
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 10, October 2026

DOI: 10.17148/IJIREEICE.2026.141024

A similar operation to Figure 5 is presented in Figure 6. The only difference is that
Figure 6 presents tilting to the right for the breadboard surface. Right mercury switch and corresponding right electromagnet are immediately activated in order to restore previous horizontal balance of the breadboard as shown in Figure 6a. Simultaneously both half right of board LEDs and three right seven-segment displays are ON, providing an indication of the above tilt. If we intentionally maintain the tilting of the breadboard preventing it from becoming horizontal, then we simulate a malfunction of electromagnets or other part of our system which could cause tilting maintenance. In this case, similarly to Figure 5b, red LED alarm is activated by sending from FPGA I/O pins logic 1 output to external red LED when critical time period value of 3 seconds is reached. As mentioned above this time value could be changed by programmer, depending on the application using our system.



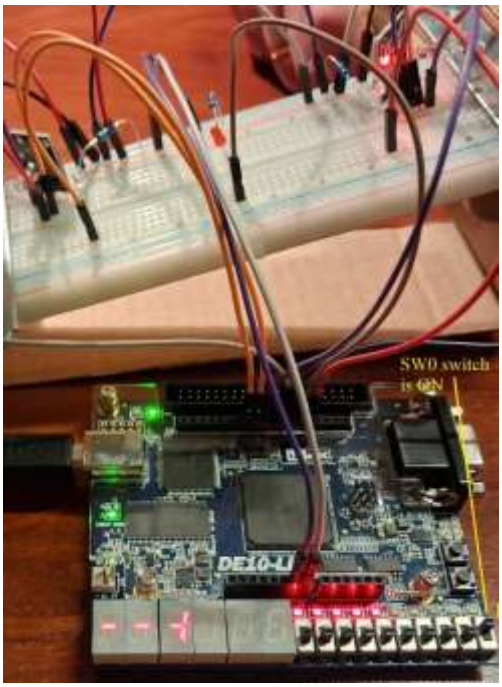
a)



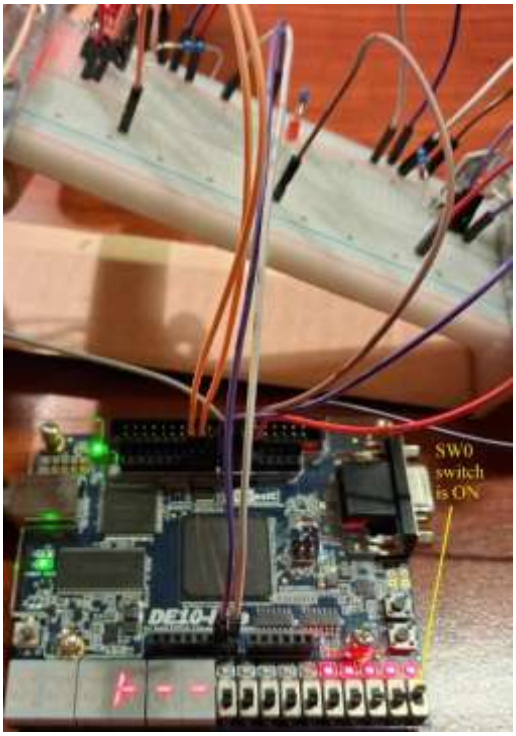
b)

Figure 6: The surface tilt monitoring and controlling system in operation mode: a) breadboard surface is tilt to the right, thus right mercury switch is activated leading to right electromagnet activation (its white LED is ON). Simultaneously half right of board LEDs and three right seven-segment displays present the above tilt. b) breadboard surface is mechanically forced to remain tilted with consequent red LED alarm activation after 3 seconds.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 10, October 2026
DOI: 10.17148/IJIREEICE.2026.141024



a)



b)

Figure 7: The surface tilt monitoring and controlling system in operation mode. SW0 board switch is ON, thus no electromagnets are activated and also time measurement process for starting red LED alarm operation, is inactive. A) surface tilt to the left with both left half of board LEDs and seven-segment displays activated, b) surface tilt to the right with both right half of board LEDs and seven-segment displays activated.

Figure 7 shows our system in operation, but there is a significant difference here compared to previous operation conditions shown in Figures 5 and 6. The difference is that FPGA board switch SW0 being OFF in previous Figures, now it is turned ON. We could call this condition an intentionally platform unloading mode. This is due to subsequent processes running simultaneously with turning SW0 to ON state. The first is the blocking of electromagnets operation although surface tilt is detected. This is obvious in both Figures 7a and 7b because in spite of breadboard tilt left (7a) or

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 10, October 2026

DOI: 10.17148/IJIREEICE.2026.141024

right (7b), respectively, with subsequent mercury tilt switches modules red lights turned to OFF state proving tilting existence, corresponding electromagnets operation white external LEDs are OFF, meaning that electromagnets are not operating. The second process that it is blocked upon turning SW0 board switch ON, is time measuring process. This means that tilting of surface will remain as long as user wants to keep it, in order to unload for example an industrial platform or a laboratory used surface. It can also be seen in both Figures 7a and 7b that FPGA's board LEDs and seven-segment displays operation, is not affected by the use of SW0 switch in the ON state.
We must mention that another important fact of our system's design is that it receives input mercury tilt sensor switches values periodically, ensuring continuous monitoring of surface tilting.

PROGRAMING THE SYSTEM

We used Quartus Prime Lite Edition 21.1.1 to create the VHDL programs of our system. A flowchart diagram, presenting main functions of our system is presented in Figure 8, while the APPENDIX contains the whole VHDL program.
It is clear that the system operates five basic functions. All of them use processes in VHDL programming language. These processes are running as long as the system is ON. All external and FPGA's board LEDs as well as electromagnets, are programmed to work as logic outputs, thus they are at the ON state if they receive binary '1' as logic output from FPGA's I/O pins. Mercury tilt switches and SW0 board switch act as logic inputs, thus they are sending binary '1' into FPGA's I/O pins upon start operating.

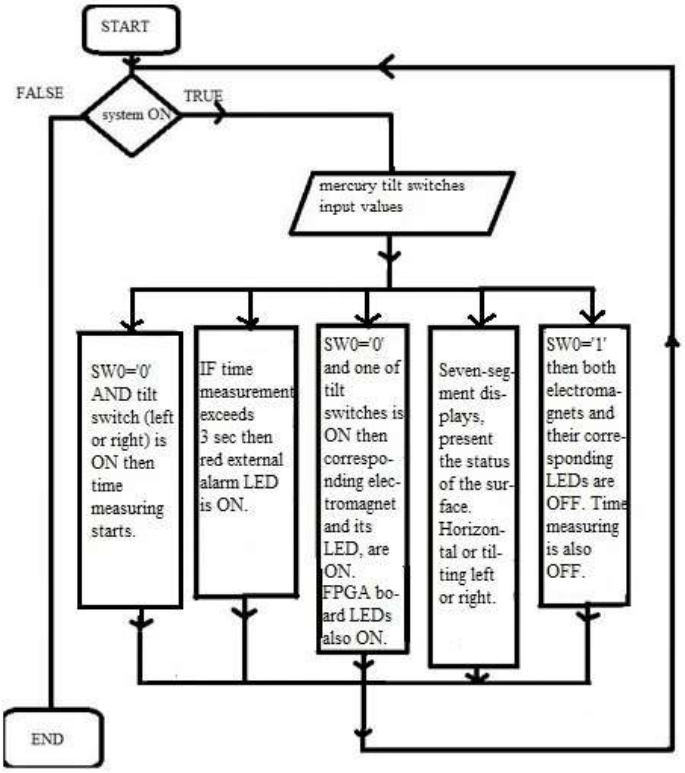


Figure 8: Flowchart diagram, presenting main functions-processes of our system.

First function of Figure 8 is associated with time measuring process. It demands that both SW0 board switch is OFF and surface tilt to the left or right, occurs. Then we get a simultaneous start of time measuring in order to activate red external LED alarm in case that set time period value of 3 seconds remaining tilt, is reached or exceeded. In this process, as it can be seen in the APPENDIX, we set seconds value to zero upon previous horizontal balance is achieved. This helped us to avoid using complex IF THEN ELSE structures and also provided accurate time measurements.

The second function is involved exclusively with tilting time period value, checking whether it was reached or exceeded and simultaneously activate external red LED alarm. Depending on the application that our system could be used, this red LED alarm represents an additional safety system able to be activated, in order to overcome malfunction of electromagnets which prevented surface transition to initial horizontal balance.

DOI: 10.17148/IJIREEICE.2026.141024

Third function is associated with tilting process in case that SW0 board switch is OFF. When surface tilting is detected by one of mercury switches binary '1' is sent to FPGA and corresponding electromagnet is activated with binary '1' sent from FPGA, resulting in magnetic force effect on breadboard's metal strip, thus restoring previous horizontal balance of breadboard surface. Simultaneously with electromagnet activation we send logic '1' to corresponding electromagnet white LED, in order to have additional evidence of its appropriate function. Tilting condition and direction is indicated by lighting half left or half right FPGA board LEDs by using a relative logic output signal in VHDL program.
Fourth function is related to seven-segment displays of the FPGA board and their role in presenting the status of surface tilting. If no tilting occurs surface horizontal balance is indicated in seven-segment displays by showing six little lines one in each display

(– – – – – –), as it was mentioned in Figure 4. When left tilting occurs three left displays present the situation, while three right displays present right surface tilting, as shown in Figures 5,6 and 7.
Finally fifth function is related to SW0 board switch transition to ON state. If this transition takes place then we programmed the system to have its electromagnets and their operating LEDs inactive. This means that if the surface is intentionally tilted it will remain in this mode until user transits SW0 to OFF state, leading to electromagnets activation. Additionally the process of time measuring is programmed to be also inactive as long as SW0 board switch remains to logic '1'.
The flexibility of our VHDL program makes it suitable for a variety of industrial or laboratory applications.

CONCLUSION

An FPGA-based, surface monitoring and controlling system is presented in this work. Our system is able to continuously monitor surface tilting using two mercury tilt sensor switches. It is also able to restore surface previous horizontal balance when tilting occurs, by using two electromagnets exerting magnetic force on metal strips existing on surface's edges. Operation of electromagnets is additionally indicated by using two external white LEDs activated upon electromagnets activation. Our system can also provide a time measuring capability for checking whether surface tilt lasts more than the value set by programmer. In this case a red LED alarm system is activated indicating possible electromagnets malfunction. The system also provides user with the ability of maintaining surface tilt as long as SW0 board switch is at the ON state, by inactivating electromagnets and time measuring processes. All surface tilting modes horizontal, left or right are appropriately displayed on FPGA board seven-segment displays.

The system is easy to be manufactured, providing also the benefit of low cost. Additionally, input tilt values could be sent to an AI system to achieve simultaneous monitor and control of a large number of surfaces and platforms especially in the case of an industrial or laboratory environment.

REFERENCES

1. M. Yussup, M.M. Ibrahim, L. Lombigit, N.A.A. Rahman and M.R.M. Zin, "Implementation of data acquisition interface using on-board field-programmable gate array (FPGA) universal serial bus (USB) link", Advanc. In Nuclear Research and Energy Development, AIP Conf. Proc. 1584, 69-72 (2014), doi: 10.1063/1.4866106
2. A. Gujar, International Journal of Computer Science and Information Technologies, "Image Encryption using AES Algorithm based on FPGA", vol 5, (5), 2014, 6853-6859
3. S. Singh, A.K. Saini, R. Saini, I.J. Image, Graphics and Signal Processing, "Interfacing the Analog Camera with FPGA Board for Real-time Video Acquisition" 2014, 4, 32-38, DOI: 10.5815/ijigsp.2014.04.04
4. S. Muthukrishnan and R. Priyadharsini, International Journal of Computer Science and Mobile Computing, "32-Bit RISC and DSP System Design in an FPGA" vol 3, issue 12, Dec. 2014, pg. 361-368
5. L. Chen, Y. Chang, L. Yan, IEEE Transactions on Geoscience and Remote Sensing, "On-orbit real-time variational image destriping: FPGA architecture and implementation", 10.1109/tgrs.2022.3140428, 2022, pp. 1-1
6. S. Yarlagadda, S. Kaza, A. Tummala, E. Babu, R. Prabhakar, Information Technology in Industry, "The reduction of Crosstalk in VLSI due to parallel bus structure using Data Compression Bus Encoding technique implemented on Artix 7 FPGA Architecture", 10.17762/itii.v9i1.151, 2021, Vol 9 (1), pp. 456-460
7. C. Du, Y. Yamaguchi, Electronics, "High-Level Synthesis Design for Stencil Computations on FPGA with High Bandwidth Memory", 2020, 9(8), 1275; https://doi.org/10.3390/electronics9081275
8. X. Hao, C. Lin and Q. Wu, Electronics, "A Parallel Timing Synchronization Structure in Real-Time High Transmission Capacity Wireless Communication Systems", 2020, 9(4), 652; https://doi.org/10.3390/electronics9040652

DOI: 10.17148/IJIREEICE.2026.141024

9. P. A. Bawiskar, R.K. Agrawal, International Journal of Innovative Research in Science, Engineering and Technology, "FPGA Based Home Security System" vol.4, issue 12, Dec. 2015, p. 12865-12869, DOI: 10.15680/IJIRSET.2015.0412139
10. K. Saroch, A. Sharma, IOSR Journal of Electronics and Communication Engineering, "FPGA Based System Login Security Lock Design Using Finite

State Machine" vol 5, issue 3, Mar.-Apr. 2013, pp 70-75
11. R.S. Parikh, Int. Journal of Engineering Research and Application, "Alarm System Implementation on Field Programmable Gate Array" vol 8, issue 1, Jan. 2018, pp 01-04.
12. E. I. Dimitriadis and L. Dimitriadis, "A Simple, Low Cost and Multiple Input Alarm System, Functioning as Finite State Machine (FSM), Using VHDL and FPGAs", Journal of Active and Passive Electronic Devices, vol. 17, pp. 307–315, 2024.
13. E. I. Dimitriadis and L. Dimitriadis, "A g-sensor based alarm system, for multiple tilt sensor applications, using VHDL and FPGAs", Journal of Active and Passive Electronic Devices, vol. 18, pp. 119-129, 2024.
14. Evangelos I. Dimitriadis, Leonidas Dimitriadis and Aristeidis Grigoriadis, "A touch-sensing system for precise robotic arm control, using force-sensing resistors (FSR), VHDL and FPGAs", International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, vol. 13, issue 7, pp. 36-50, July 2025.
15. Leonidas Dimitriadis and Evangelos I. Dimitriadis, "A Novel System for Monitoring and Controlling Industrial Engine Operation, using Vibration, Magnetic Field, Humidity and Temperature Sensors, Implemented with FPGAs and VHDL", International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, vol. 14, issue 5, pp. 1-18, May 2026.
16. F. Gago, L.Rodríguez-Ramos, J. J. Díaz, "Tip-Tilt Mirror Control Based on FPGA for an Adaptive Optics System", 3rd Southern Conference on Programmable Logic, 2007, DOI:10.1109/SPL.2007.371718, Corpus ID: 9008342
17. Artur Zawadzki, Marek Gorgoń, "Automatically controlled pan–tilt smart camera with FPGA based image analysis system dedicated to real-time tracking of a moving object", Journal of Systems Architecture, Date: November 2015, Pages: 681-692, Volume: Volume 61, Issue 10
18. Xin Xu, Zheng Chen, Hong Li, Shigui Ma, Liheng Wu, Wenbo Wang, Yunkai Dong and Weiwei Zhan, "Design of A High-Precision Component-Type Vertical Pendulum Tiltmeter Based on FPGA", Sensors, 2023, 23(18), 7998, https://doi.org/10.3390/s23187998

APPENDIX

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.std_logic_signed.ALL;
use ieee.std_logic_unsigned.ALL;
entity DE10_Lite_mercury_tilt_switches_electromagnets is
generic(ClockFrequencyHz : integer:=50000000);
port
(
rst : in std_logic; --SW8
nRst    : in std_logic; -- Negative reset
merc_switch_left : in std_logic;
merc_switch_right : in std_logic;
electromagnet_left : out std_logic;
electromagnet_right : out std_logic;
electromagnet_left_oper : out std_logic;
electromagnet_right_oper : out std_logic;
Seconds : inout integer;
Ticks : inout integer;
red_led_alarm : out std_logic;
SW0 : in std_logic;
led0_out: out std_logic;
led1_out: out std_logic;
led2_out: out std_logic;
led3_out: out std_logic;
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 10, October 2026
DOI: 10.17148/IJIREEICE.2026.141024

```
led4_out: out std_logic;
led5_out: out std_logic;
led6_out: out std_logic;
led7_out: out std_logic;
led8_out: out std_logic;
led9_out: out std_logic;
-- Clocks
ADC_CLK_10: in std_logic;
MAX10_CLK1_50: in std_logic;
MAX10_CLK2_50: in std_logic;
-- KEYs
KEY: in std_logic_vector(1 downto 0);
-- HEX
HEX0: out std_logic_vector(7 downto 0);
HEX1: out std_logic_vector(7 downto 0);
HEX2: out std_logic_vector(7 downto 0);
HEX3: out std_logic_vector(7 downto 0);
HEX4: out std_logic_vector(7 downto 0);
HEX5: out std_logic_vector(7 downto 0);
ARDUINO_IO: inout std_logic_vector(15 downto 0);
ARDUINO_RESET_N: inout std_logic);
end entity;
architecture DE10_Lite_mercury_tilt_switches_electromagnets_Arch of
DE10_Lite_mercury_tilt_switches_electromagnets is
-- Analog to Digital Converter IP core
component myADC is
port
(
clk_clk: in std_logic := 'X';
modular_adc_0_command_valid: in std_logic := 'X';
modular_adc_0_command_channel: in std_logic_vector(4 downto 0) := (others => 'X');
modular_adc_0_command_startofpacket: in std_logic := 'X';
modular_adc_0_command_endofpacket: in std_logic := 'X';
modular_adc_0_command_ready: out std_logic;
modular_adc_0_response_valid: out std_logic;
modular_adc_0_response_channel: out std_logic_vector(4 downto 0);
modular_adc_0_response_data: out std_logic_vector(11 downto 0);
modular_adc_0_response_startofpacket: out std_logic;
modular_adc_0_response_endofpacket: out std_logic;
reset_reset_n: in std_logic
);
end component myADC;
signal modular_adc_0_command_valid: std_logic;
signal modular_adc_0_command_channel: std_logic_vector(4 downto 0);
signal modular_adc_0_command_startofpacket: std_logic;
signal modular_adc_0_command_endofpacket: std_logic;
signal modular_adc_0_command_ready: std_logic;
signal modular_adc_0_response_valid: std_logic;
signal modular_adc_0_response_channel: std_logic_vector(4 downto 0);
signal modular_adc_0_response_data: std_logic_vector(11 downto 0);
signal modular_adc_0_response_startofpacket: std_logic;
signal modular_adc_0_response_endofpacket: std_logic;
signal clk_clk: std_logic;
signal reset_reset_n: std_logic;
type state_machines is (sm0,sm1, sm2, sm3, sm4);
signal sm: state_machines;
-- signals to store conversion results
signal ADCIN1,ADCIN4, ADCIN3,ADCIN2: std_logic_vector(11 downto 0);
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 10, October 2026

DOI: 10.17148/IJIREEICE.2026.141024

```
signal AD1,AD4, AD3,AD2: std_logic_vector(11 downto 0);
-- signal for board and external LEDs
signal state_led_left : std_logic;
signal state_led_right : std_logic;
signal state_led_alarm : std_logic;
begin
-- ADC port map
adc1: myADC port map
(
modular_adc_0_command_valid => modular_adc_0_command_valid,
modular_adc_0_command_channel => modular_adc_0_command_channel,
modular_adc_0_command_startofpacket => modular_adc_0_command_startofpacket,
modular_adc_0_command_endofpacket => modular_adc_0_command_endofpacket,
modular_adc_0_command_ready => modular_adc_0_command_ready,
modular_adc_0_response_valid => modular_adc_0_response_valid,
modular_adc_0_response_channel => modular_adc_0_response_channel,
modular_adc_0_response_data => modular_adc_0_response_data,
modular_adc_0_response_startofpacket => modular_adc_0_response_startofpacket,
modular_adc_0_response_endofpacket => modular_adc_0_response_endofpacket,
clk_clk => clk_clk,
reset_reset_n => reset_reset_n
);
clk_clk <= MAX10_CLK1_50;
reset_reset_n <= KEY(0);
-- process for reading new samples
p1: process(reset_reset_n, clk_clk)
begin
if reset_reset_n = '0' then
        sm <= sm0;
elsif rising_edge(clk_clk) then
        case sm is
                when sm0 =>
                        sm <= sm1;
                        modular_adc_0_command_valid <= '1';
                        modular_adc_0_command_channel <= "00001";
                when sm1 =>
                        if modular_adc_0_response_valid = '1' then
                                modular_adc_0_command_channel <= "00010";
                         ADCIN4 <= modular_adc_0_response_data;
                                sm <= sm2;
                        end if;
                when sm2 =>
                        if modular_adc_0_response_valid = '1' then
                                modular_adc_0_command_channel <= "00001";
                                modular_adc_0_command_channel <= "00011";
                                ADCIN1 <= modular_adc_0_response_data;
                                sm <= sm1;
                                sm <= sm3;
                        end if;
                when sm3 =>
                        if modular_adc_0_response_valid = '1' then
                                modular_adc_0_command_channel <= "00100";
                                ADCIN2 <= modular_adc_0_response_data;
                                sm <= sm4;
                        end if;
                when sm4 =>
                        if modular_adc_0_response_valid = '1' then
                                modular_adc_0_command_channel <= "00001";
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 10, October 2026

DOI: 10.17148/IJIREEICE.2026.141024

```
                                    ADCIN3 <= modular_adc_0_response_data;
                                    sm <= sm1;
                                end if;
                            when others =>
                    end case;
end if;
end process;
process(MAX10_CLK1_50) is
 begin
IF SW0='0' AND (merc_switch_left='1' OR merc_switch_right='1') THEN
                if rising_edge(MAX10_CLK1_50) then --ισοδύναμο του IF CLK'EVENT AND CLK='1'
            ------ If the negative reset signal is active
            if nRst = '0' then
               Ticks   <= 0;
               Seconds <= 0;
            else
               -- True once every second
               if Ticks = ClockFrequencyHz - 1 then
                  Ticks <= 0;
                     Seconds <= Seconds + 1;
               else
                  Ticks <= Ticks + 1;
               end if;
            end if;
        end if;
                        else
                        Ticks   <= 0;
        Seconds <= 0;
                        end if;
    end process;
process(SW0, merc_switch_left, merc_switch_right, state_led_left, state_led_right)
begin
IF SW0 ='0' THEN
IF merc_switch_left='1' THEN
electromagnet_left <= '1';
electromagnet_left_oper<='1';
state_led_left<='1';
ELSE
electromagnet_left <= '0';
electromagnet_left_oper<='0';
state_led_left<='0';
END IF;
IF merc_switch_right='1' THEN
electromagnet_right <= '1';
electromagnet_right_oper<='1';
state_led_right<='1';
ELSE
electromagnet_right <= '0';
electromagnet_right_oper<='0';
state_led_right<='0';
END IF;
ELSE
IF merc_switch_left='1' THEN
electromagnet_left <= '0';
electromagnet_right <= '0';
electromagnet_right_oper<='0';
electromagnet_left_oper<='0';
state_led_left<='1';
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 10, October 2026

DOI: 10.17148/IJIREEICE.2026.141024

```
ELSIF merc_switch_right='1' THEN
electromagnet_right <= '0';
electromagnet_left <= '0';
electromagnet_right_oper<='0';
electromagnet_left_oper<='0';
state_led_right<='1';
END IF;
END IF;
end process;
process(state_led_alarm, seconds)
BEGIN
IF seconds >=3 THEN
state_led_alarm <= '1';
ELSE
state_led_alarm <= '0';
END IF;
END PROCESS;
red_led_alarm <= state_led_alarm;
led0_out<=state_led_right;
led1_out<=state_led_right;
led2_out<=state_led_right;
led3_out<=state_led_right;
led4_out<=state_led_right;
led5_out<=state_led_left;
led6_out<=state_led_left;
led7_out<=state_led_left;
led8_out<=state_led_left;
led9_out<=state_led_left;
process(merc_switch_left, merc_switch_right)
begin
IF (merc_switch_left='1' AND merc_switch_right='0') THEN
HEX0 <= "11111111";
HEX1 <= "11111111";
HEX2 <= "11111111";
HEX3 <= "10111001";
HEX4 <= "10111111";
HEX5 <= "10111111";
ELSIF (merc_switch_right='1' AND merc_switch_left='0') THEN
HEX0 <= "10111111";
HEX1 <= "10111111";
HEX2 <= "10001111";
HEX3 <= "11111111";
HEX4 <= "11111111";
HEX5 <= "11111111";
ELSIF (merc_switch_left='0' AND merc_switch_right='0') THEN
HEX0 <= "10111111";
HEX1 <= "10111111";
HEX2 <= "10111111";
HEX3 <= "10111111";
HEX4 <= "10111111";
HEX5 <= "10111111";
end if;
end process;
end architecture;
```