12th INTERNATIONAL SCIENTIFIC CONFERENCE ON DEFENSIVE TECHNOLOGIES
OTEH 2026
Serbia, Tara, 21 - 24 October 2026

IMPLEMENTATION AND EXPERIMENTAL VERIFICATION OF ENVIRONMENTAL SAFETY SENSOR DEVICE FOR FUZES IN AERIAL BOMBS

SLOBODAN PELEMIŠ
Singidunum University, Vlatacom Institute, Belgrade, slobodan.pelemis@vlatacom.com

NIKOLA LATINOVIĆ
Vlatacom Institute, Belgrade, nikola.latinovic@vlatacom.com

MLADEN VEINOVIĆ
Singidunum University, Belgrade, mveinovic@singidunum.ac.rs

Abstract: *In this paper, a methodology for the implementation of an environmental safety sensor device as part of a fuzing system for aerial bombs will be presented. Complete hardware and software for this system architecture will be described herein.*
Environmental sensors have a wide range of applications in industry, military technologies, IoT, and many more. These are usually devices that can measure various physical parameters of the environment and convert data into electrical signals suitable for use with electronic devices. In military applications, in weapon systems, the most important factor is safety during the handling and use of explosive products. In this particular case, the application of the environmental sensors in the fuzing systems for aerial bombs is crucial for the safety of pilots and aircrafts.
The test results of the implemented device will be shown as proof of functionality under the given conditions. Software is written in C language as the operating system of a microcontroller used as the main control unit of the entire system.
The core contribution of this work is the analysis of safety features of a device that generates the appropriate signals as conditions for the aerial bomb fuze's performance of its arming process.

Keywords: *safety sensor, environmental conditions, electronics, aerial bomb, C language.*

1. INTRODUCTION

Aerial bombs detonation is controlled by the action of a fuze. A fuze is a device that causes the detonation of an explosive charge at the proper time after certain conditions are met. Part of the fuzing system is also a product called initiator mounted in the middle of the bomb between lugs as shown in Figure 1. [1]

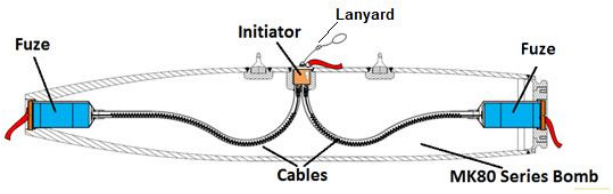


Figure 1. Position of the fuzes and initiators in aerial bomb

The initiator is a cylindrically shaped metal component that is installed in the bomb charging well.
The initiator contains two essential components:

- ***Launch sensor,*** MP1 (switch): which in the active position (closed) connects the battery to the electronic block of the fuze and to the initiator;
- ***Environmental sensor*** with appropriate comparator. This component provides data, designated as MP2, about the pressure of the air stream, indicating that the aircraft has released the bomb and the bomb is flying to the target.

A dedicated hardware-software architecture has been developed for the realization of the proposed sensor device for safety in air-bomb fuzing systems.
The system relies on the integration of a microcontroller unit (MCU) and a differential pressure sensor to ensure that the arming signal is generated only after meeting the given environmental conditions.

Environmental sensor does not serve to activate the bomb on the target, but as a safety device that prevents the fuze from operating until the bomb has safely left the vicinity of the aircraft. [1], [2]

Safety role means that the sensor will generate arming signal when the pressure of air stream is greater than 0.2 psi. This signal is monitored by MCU in the electronic block of the fuze during the bomb flying.

Advantage of this integrated software in a relation with pressure sensor is in the following: when we are comparing system used in AUFK M-91 fuze, with system presented in this paper, we can see that AUFK M-91 fuze

Corresponding author: S. Pelemiš;
E-mail: slobodan.pelemis@vlatacom.com
DOI: 10.5937/OTEH26022P

in order to be armed, must meet the electrical condition, i.e. to receive a certain electric pulse of nominal value of 24V from the plane. It represents a safety requirement and as such would not be acceptable to the modern 1316 safety standard used today. [3]

In this paper, by introducing an initiator based on the differential pressure sensor, we have two safety conditions instead of one. First one represents the opening of the switches when dropping the bomb in form of MP1 signal. Second one is a physical component, i.e. the measurement of differential pressure, which is directly proportional to the speed of the air flow, in real time conditions shown as MP2 signal.

This creates two independent safety conditions that form the basis for the fulfillment of standard 1316.

2. HARDWARE DESIGN AND SIGNAL CONDITIONING

Environmental pressure sensor is located inside the steel housing of the initiator and is activated during the flight of the aerial bomb.

Electronic assembly of complete system which includes mentioned sensor is shown in **Figure 2**.
Characteristics of sensor are given in **Table 1**.

Table 1. Pressure sensor characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Pressure Range	P_{OP}	0	-	50	kPa
Supply Voltage	V_S	4.75	5.0	5.25	V_{dc}
Supply Current	I_O	-	7.0	10	mAdc
Min. Pressure Offset (0 °C to 85 °C), Vs=5 V	V_{off}	0.088	0.2	0.313	Vdc
Full Scale Output (0 °C to 85 °C), Vs=5 V	V_{FSO}	4.587	4.7	4.813	Vdc
Full Scale Span (0 °C to 85 °C), Vs=5 V	V_{FSS}	-	4.5	-	Vdc
Accuracy (0 °C to 85 °C)	-	-	-	±2.5	$\%V_{FSS}$
Sensitivity	V/P	-	90	-	mV/kPa
Response Time	t_R	-	1.0	-	ms
Output Source Current at Full Scale Output	I_{o+}	-	0.1	-	mAdc
Warm-up Time	-	-	20	-	ms
Offset Stability	-	-	±0.5	-	$\%V_{FSS}$

Upper PCB contains differential pressure sensor and ESD protection diodes.

Lower PCB contains pressure switches which are responsible for providing battery power supply to the fuze electronic block.

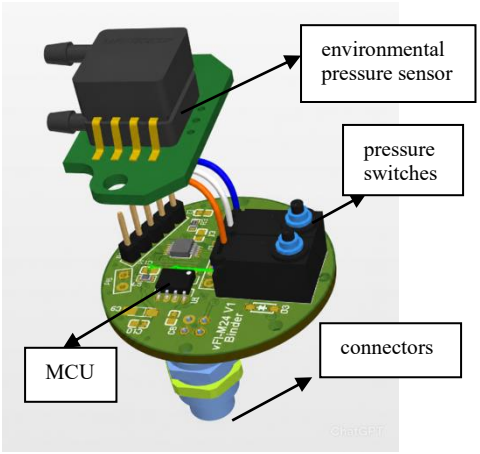


Figure 2. Electronic PCBs of the initiator

The system relies on the integration of a microcontroller unit (MCU) [4] and a differential pressure sensor to ensure that the arming signal is generated only after meeting the given environmental conditions.
Electrical block scheme of complete system is shown in Figure 3**.** [5]

The main purpose of the system is to measure differential pressure, process the sensor signal inside the MCU, and generate a digital output signal when the measured pressure exceeds a predefined Threshold. [6]

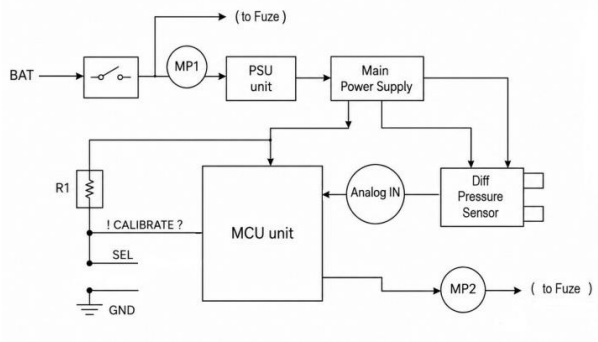


Figure 3. Electrical block scheme of complete system

The system is powered by the battery input (**BAT**). After passing through the main power switch, the supply voltage is available at signal **MP1** which is first signal to be achieved. **PSU Unit** represents preliminary power conditioning stage. Main Power Supply generates the regulated voltages required by the MCU and the differential pressure sensor.

The regulated outputs from the Main Power Supply provide stable operating voltages for the MCU unit and the differential pressure sensor. This arrangement ensures reliable operation regardless of battery voltage variations.

The Differential Pressure Sensor measures the pressure difference between its two pressure ports.

The sensor produces an analog voltage proportional to the measured differential pressure. This analog signal is routed to the MCU through the Analog IN interface.

The MCU continuously samples this signal using its internal Analog-to-Digital Converter (ADC), converts the analog voltage into a digital value, and evaluates the measured pressure against predefined limits.

The calibration circuitry consists of:

- R1 pull-up resistor;
- CALIBRATE input;
- SEL input;
- Ground reference.

These signals allow the system to enter a calibration mode during startup.

When the CALIBRATE line is active, the MCU performs a sensor offset calibration procedure and stores the calculated offset values into non-volatile memory. This compensates for sensor tolerances, temperature effects, and manufacturing variations. [7]

Regarding the output section, it is important to note that the **MCU** generates a digital output signal that is available at MP2, which indicates wheather the measured pressure exceeds the programmed Threshold.

3. SOFTWARE DESIGN AND IMPLEMENTATION

The software support, implemented in the C language, directly controls the states of the hardware components shown in the schematic on **Figure 3**. [8]

Sensor expected measurement range indicated in the experimental results equates to the value of V_{off} + 620.7 mV ± 5% of measurement inaccuracy. The value of 620.7 mV is derived from the formula presented by the device vendor:

$$V_p = V_{off} + \frac{90}{0.145} \cdot P\ [mV] \qquad (1)$$

where V_p is the output voltage in [mV], V_{off} is voltage offset in [mV], and P is pressure (difference) in [psi] which produces the output voltage V_p.

3.1 SOFTWARE CODE WRITTEN IN C LANGUAGE

Following lines show the code written for check of the 0.2 psi pressure value of the sensor. [9], [10]

```
/*
 * Pressure sensor equation: Vp-Voff = (90/0.145)*P [mV]; P is in Psi units
 * for 0.2Psi Vp-Voff = 124mV
 * if using 12bit A/D conversions 1LSB is 1.22mV so threshold is equal to 102
 */

#define MP2_THRESHOLD12b  102

//Functions prototypes
unsigned char MP2_Read2B (void);
void writeOffsetToMem2B (unsigned char tempHigh, unsigned char tempLow);
void MP2_Calibrate2B (unsigned char*memLow, unsigned char *memHigh);

//Main program
void main (void)
{
  unsigned char  memLow=0, memHigh=0;
  int i=0, j=0;
  P04_PUSHPULL_MODE   //P0.4 output push pull  -- MP2
  P01_INPUT_MODE;       //P0.1 input mode -- CALIBRATE
 P06_PUSHPULL_MODE;         //led on initiator board

  P04=0;
  P06=0;
  if(P01)
  {
        while(1)
        {
              if (MP2_Read2B())
              {
                    P04=1;
              }
              else
              {
                    P04=0;
              }
        }
  }
  else
  {
        P06=1;
        MP2_Calibrate2B(&memLow, &memHigh);
        writeOffsetToMem2B(memHigh, memLow);
        P06=0;
        while(1);
  }
}
```

Software support, implemented in C language, directly manages the states of the hardware components shown in the diagram in **Figure 3**.
At the very beginning of the execution of the *main()* function, the input-output pins of the microcontroller is initialized as follows:

- *Port P0.4* is configured in push-pull operation mode. According to the scheme on **Figure 3**, this pin directly controls the MP2 output signal to the fuze.
- *Port P0.1* is set to input mode of operation and corresponds to the line *!CALIBRATE?* from the electrical diagram.
- *Port P0.6* is configured as a push-pull output to control the LED on the initiator board, which serves as a visual status indicator during calibration.

After initialization, the software checks the state of pin *P0.1(if(P01))* to determine the branching of the algorithm:

- **Detection and assurance mode (Normal operation)**: If pin *P0.1* is at a high logic level (calibration switch is open), the system enters an infinite *while(1)* loop where the *MP2_Read2B()* function is continuously called. This function reads the value from the Analog IN position

(pressure sensor output). In the code, the safety threshold *MP2_THRESHOLD12b* is defined with a value of 102. This value directly corresponds to a physical pressure of 0.2 psi (that is, a voltage shift of 124 mV above the offset, taking into account the 12-bit resolution of the ADC where 1 LSB is 1.22mV). If the differential pressure read exceeds the defined threshold, the software sets pin P0.4 to a high logic level *(P04=1)*, which activates the output to the igniter via MP2. Otherwise, the output remains inactive *(P04=0)*, ensuring system security.

- **Calibration mode**: If pin *P0.1* is low at startup *(else branch)*, the LED is activated (*P06=1*). The software then calls the function *MP2_Calibrate2B (&memLow, &memHigh)* to measure the current offset of the pressure sensor at rest, and then writes these values permanently to memory via the function *writeOffsetToMem2B(memHigh, me*mLow). Upon completion of the calibration, the LED turns off (*P06=0*), and the microcontroller enters a waiting state (*while(1)*), blocking further operation of the igniter until the system is restarted in normal mode.

With this synergy of hardware lines and software conditions, a high resistance to unwanted activation of the aerial bomb was achieved, because the arming signal cannot be generated without an adequate change in the environmental pressure that follows the specific flight trajectory of the aircraft.

4. EXPERIMENTAL TESTS WITH RESULTS AND FUNCTIONALITY PROOF

In order to verify developed software which is implemented on MCU, experimental tests are performed. [11]

Equipment necessary for these tests is:

- pneumatic calibration pump with connected digital pressure gauge;
- initiator electronics test box;
- power supply 10.8 V DC;
- voltage meter.

Pneumatic calibration pump with digital pressure gauge shown in **Figure 4** is necessary for simulating air stream pressure which will occur in real-time working conditions during air bomb flight.



Figure 4. Pneumatic calibration pump with digital gauge

By using the appropriate adaptor and a hose, pressure port of the sensor of UUT and the output of pneumatic calibration pump must be connected as shown in Figure 5. UUT in this case is pressure sensor located inside of the initiator housing.

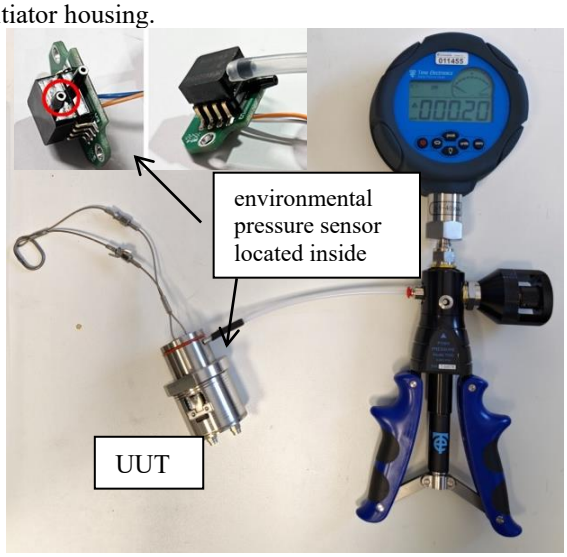


Figure 5. Pump connected with UUT sensor

First part of the test setup is shown on Figure 5. To complete test, we must have also connected power supply, voltage meter and initiator test box. Initiator test box is used to show when MP1 and MP2 signals are activated. Rest of the setup we can see on Figure 6.

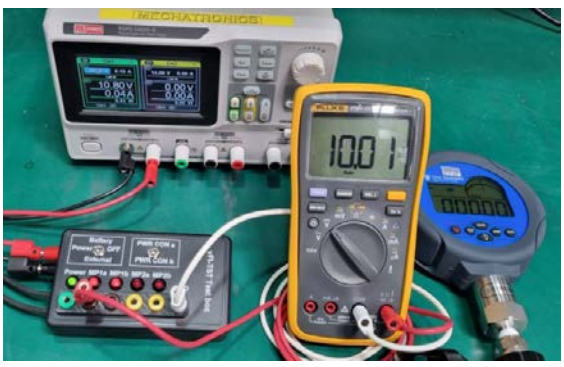


Figure 6. MP1 signals activated

In above Figure, it is demonstrated that with air stream pressure lower than 0.2 psi (see display on calibration pump), we have only MP1 signals active (MP1 diodes

glows red, and MP2 diodes don't). This means that when air bomb is released on active mode, switch MP1 is closed, the battery connects to electronic block and arming process started. Environmental sensor still didn't detect air stream and didn't produce any analogue output signal. This is implemented through the C language written code.

Next step is to produce pressure of 0.2 psi or bigger which happens during bomb flight. [11]

We can see pressure value manually defined on pump on **Figure 7**.



Figure 7. MP2 signals activated

When the pressure of the air stream reaches 0.2 psi or above, pressure sensor produce signal MP2=TRUE. This signal is monitored by the microcontroller in the electronic block of the fuze during the bomb moving. The microcontroller starts monitoring this signal when 0.9 of the arming time has elapsed, i.e. at the beginning of the last 10% of the arming time. The sensor will generate an MP2 signal TRUE only if the pressure of air stream is greater than 0.2 psi. (for example if the arming time is preset at 10 seconds, in order for the fuze to arm, this presure must be greater than 0.2psi after 9 and before 10 seconds of arming time have elapsed). **Figure 7** shows that, with the appropriate pressure applied, beside the MP1 diodes, the MP2 diodes on the test box also glow, which indicates the start of the arming process. [11]
In order to prevent influence of noise and interference the microcontroller checks signal MP2 three times after the completion of the 90% of arming time. In case the signal is present during all of these checks, microcontroler concludes that the bomb is moving away and distance from the aircraft is safe enough for the arming process of the fuze to be completed.. Conversely, if the MP2 signal is not TRUE during the last 10% of arming time (3 consecutive readings), arming process will be stopped and arming time resets. The fuze will not be armed.

5. CONCLUSION

This paper has presented the implementation and experimental verification of an environmental safety sensor device designed for aerial bomb fuzing systems. By integrating a microcontroller unit with a differential pressure sensor, the developed hardware-software architecture successfully addresses the strict safety demands critical to military applications.

The embedded software, written in C language, provides reliable signal processing and introduces a dedicated calibration routine that effectively compensates for sensor tolerances and environmental variations.

Through the experimental laboratory setup which includes a pneumatic calibration pump and a dedicated test box, we successfully validated the system's functionality under simulated flight conditions. The testing confirmed that the crucial arming signal (MP2) triggers only when the airflow pressure safely crosses the 0.2 psi threshold.

It is shown that comparing to other similar products used in military, we have two safety conditions ensured (explained through MP1 and MP2 signals) which is advantage regarding the modern military standard 1316. Other fuzes such as for example AUFK type has one condition in form of electrical impulse variable.

Ultimately, the core value of this design lies in its contribution to pilot and aircraft safety. By tying the arming sequence directly to a precise, verifiable change in environmental pressure along the flight trajectory, the system achieves excellent resistance to accidental or premature activation.

References

[1] U.S. DEPARTMENT OF DEFENSE, MIL-STD-1316F: *Safety Criteria for Fuze Design*, Washington, DC, USA – DoD, 2017.

[2] DEURA,N., DOMAZET,D., NIJEMČEVIĆ,S.: *Fuze vAF-M17 Electronic block test software*, in Proc. 10th Scientific Conf. Defensive Technologies (OTEH), Belgrade, Serbia, pp. 121-127.

[3] KRUŠIK,H.,K.,*Upaljači,* https://krusik.rs/images/pdf/UPALJACI.pdf, page 26

[4] PEPLOW,R., SHENOUDA,D.: *Digital System Design – Use of Microcontroller,* River Publishers

[5] OH,J.S., HAN,Y.M., CHOI,S.B., NGUYEN,V.Q., MOON, S.J.: *Design of a one chip board microcontrol unit for active vibration control of a naval ship mounting system*, IOP Publish. doi: 10.1088/0964176/21/8/087001.

[6] ŽIVKOVIĆ,M., DOMAZET,D., LATINOVIĆ,N.: *Fuze vAF – M17 microcontroller software*, in Proc. 10th Int. Scientific Conf. Defensive Technologies (OTEH), Belgrade, Serbia, 2022, pp. 114-120.

[7] TANG, M., ZHANG, X., LIU, Y., JI, P., FAN,L., TAN,Q., LIANG, T.: *Design and preparation of a high immunity piezo resistive sensor*, Journal of Measurement Science and Instrumentation, Vol.17 No.1 Mar. 2026, doi: 10.62756/jmsi. 1674-8042.2026010.

[8] XUEJIAN, L.: *Design of Safe C language for program verification*, IEEE Access, doi: 10.1109/ACCESS.2026.3656523.

[9] ZHAO,P., ZHANG,Y., CHEN,Y., JIA,Y.: *Modular Programming in C for MCU,* doi: 10.1109/EIECS66708.2025.1128, 2025 5th International Conference on Electronic Information Engineering and Computer Science

[10] KERNIGHAN,B.W., RITCHIE,D.M.: *The C programming language*, Second Edition, Prentice Hall Software Series.

[11] VLATACOM, I.: *159_5_1.3D_Fuze vAF-M17 and Initiator vFI-M17 PCB Testing,* (Internal), 2025, unpublished.