DAFTAR LAMPIRAN

LAMPIRAN A

No. Dok. : F-PBM-16 | Tgl. Berlaku : 13 Desember 2010 | No. Rev. : 00

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id

KESEPAKATAN BIMBINGAN LAPORAN AKHIR (LA)

Kami yang bertanda tangan di bawah ini,

Pihak Pertama

Nama	: Natasya Ramadhani
NPM	: 061930322852
Jurusan	: Teknik Elektro
Program Studi	: DIII Teknik Elektronika

Pihak Kedua

Nama	: Yudi Wijanarko, S.T., M.T.
NIP	: 196705111992031003
Jurusan	: Teknik Elektro
Program Studi	: DIII Teknik Elektronika

Pada hari ini Selasa tanggal 15 Februari 2022 telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir.

Konsultasi bimbingan sekurang-kurangnya 1 (satu) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari Kamis pukul 09.00 , tempat di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Laporan Akhir.

Palembang,

Pihak Pertama,

(Natasya Ramadhani)
NIM. 061930322852

Pihak Kedua,

(Yudi Wijanarko, S.T., M.T.)
NIP. 196705111992031003

Mengetahui,
Ketua Jurusan

(Ir. Iskandar Lutfi, M.T.)
NIP. 196501291991031002

LAMPIRAN B

No. Dok. : F-PBM-16 | Tgl. Berlaku : 13 Desember 2010 | No. Rev. : 00

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id

KESEPAKATAN BIMBINGAN LAPORAN AKHIR (LA)

Kami yang bertanda tangan di bawah ini,

Pihak Pertama

Nama	: Natasya Ramadhani
NPM	: 061930322852
Jurusan	: Teknik Elektro
Program Studi	: DIII Teknik Elektonika

Pihak Kedua

Nama	: Masayu Anisah, S.T., M.T.
NIP	: 197012281993032001
Jurusan	: Teknik Elektro
Program Studi	: DIV Sarjana Terapan

Pada hari ini Jumat tanggal 11 Maret 2022 telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir.

Konsultasi bimbingan sekurang-kurangnya 1 (satu) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari Kamis pukul 14.00 , tempat di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Laporan Akhir.

Palembang,

Pihak Pertama,

(Natasya Ramadhani)
NIM. 061930322852

Pihak Kedua,

(Masayu Anisah, S.T., M.T.)
NIP. 197012281993032001

Mengetahui,
Ketua Jurusan

(Ir. Iskandar Lutfi, M.T.)
NIP. 196501291991031002

LAMPIRAN C

No. Dok. : F-PBM-17 Tgl. Berlaku : 13 Desember 2010 No. Rev. : 00

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
RISET DAN TEKNOLOGI
DIREKTORAT PENDIDIKAN TINGGI VOKASI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id

LEMBAR BIMBINGAN LAPORAN AKHIR

Nama	:	Natasya Ramadhani
NIM	:	061930322852
Jurusan/Program Studi	:	Teknik Elektro/DIII Teknik Elektronika
Judul Laporan Akhir	:	Rancang Bangun Simulator *Engine Fire* pada Pesawat DC-9 Berbasis *Internet of Things* (IoT)
Pembimbing I	:	Yudi Wijanarko, S.T., M.T.

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	17/01/22	Judul & Pengajuan Proposal LA.	[signature]
2.	27/01/22	Progres Proposal LA, Rapikan, Tulis	[signature]
3.	7/2/22	Perbaiki proposal, [illegible]	[signature]
4.	15/2/22	Finalisasi Proposal, Lanjut LA.	[signature]
5.	11/3-22	[illegible] pengukuran & [illegible].	[signature]
6.	10/6-22	Progres Alat & tulis LA	[signature]
7.	21/6-22	Perbaiki alat & Lanjutkan LA.	[signature]
8.	11/7-22	Lengkapi dokumen & Laporan	[signature]
9.	12/7'22	Perbaiki LA, Lengkapi dokumen	[signature]
10.	13/7'22	Lengkapi Data & Video LA.	[signature]
11.	14/7'22	Perbaiki & Lengkapi LA & [illegible]	[signature]
12.	15/7'22	Persiapkan [illegible] ujian LA. [illegible]	[signature]

Palembang, 18/7 2022

Ketua Jurusan/KPS,

(Ir. Iskandar Lutfi, M.T.)
NIP 196[illegible]01291991031002

Catatan:
*) melingkari angka yang sesuai.
Ketua Jurusan/KPS harus memeriksa jumlah pelaksanaan bimbingan sesuai yang dipersyaratkan dalam Pedoman Laporan Akhir sebelum menandatangani lembar bimbingan ini.
Lembar pembimbingan LA ini harus dilampirkan dalam Laporan Akhir.

LAMPIRAN D

No. Dok. : F-PBM-17 | Tgl. Berlaku : 13 Desember 2010 | No. Rev. : 00

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id

LEMBAR BIMBINGAN LAPORAN AKHIR

Nama : Natasya Ramadhani
NIM : 061930322852
Jurusan/Program Studi : Teknik Elektro/DIII Teknik Elektronika
Judul Laporan Akhir : Rancang Bangun Simulator *Engine Fire* pada Pesawat DC-9 Berbasis *Internet of Things* (IoT)
Pembimbing II : Masayu Anisah, S.T., M.T.

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	22/1 2022	Outline progress proposal LA/TA, paparan judul, opsi, dan solusi.	
2.	9/2 2022	Revisi proposal LA	
3.	14/2 2022	Pengajuan proposal LA	
4.	11/3 2022	Diskusi mengenai desain alat	
5.	25/3.2022	Diskusi Bab 2 & Bab 3	
6.	14/4 - 2022	Perbaiki Bab 3 sesuai hasil diskusi	
7.	8/6 - 2022	Acc Bab 3, tambahkan Pembuatan untuk pengujian di Bab 4	
8.			
9.	18/6 - '22	Pengujian Alat, disempurnakan seminar	
10.	15/7 - 2022	Direkomendasikan Uji LA	
11.			
12.			

Palembang, 18/7 2022

Ketua Jurusan/KPS,

(Ir. Iskandar Lutfi, M.T.)
NIP 196501291991031002.

Catatan:
*) melingkari angka yang sesuai.
Ketua Jurusan/KPS harus memeriksa jumlah pelaksanaan bimbingan sesuai yang dipersyaratkan dalam Pedoman Laporan Akhir sebelum menandatangani lembar bimbingan ini.
Lembar pembimbingan LA ini harus dilampirkan dalam Laporan Akhir.

LAMPIRAN E

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
DIREKTORAT PENDIDIKAN TINGGI VOKASI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id

REKOMENDASI UJIAN LAPORAN AKHIR (LA)

Pembimbing Laporan Akhir memberikan rekomendasi kepada,

Nama	:	Natasya Ramadhani
NIM	:	061930322852
Jurusan/Program Studi	:	Teknik Elektro/DIII Teknik Elektronika
Judul Laporan Akhir	:	Rancang Bangun Simulator *Engine Fire* pada Pesawat DC-9 Berbasis *Internet of Things* (IoT)

Mahasiswa tersebut telah memenuhi persyaratan dan dapat mengikuti Ujian Laporan Akhir (LA) pada Tahun Akademik 2021/2022

Palembang, 15 Juli 2022

Pembimbing I,

(Yudi Wijanarko, S.T., M.T.)
NIP 196705111992031003

Pembimbing II,

(Masayu Anisah, S.T., M.T.)
NIP 197012281993032001

LAMPIRAN F

No. Dok. : F-PBM-23 | Tgl. Berlaku : 13 Desember 2010 | No. Rev. : 00

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
DIREKTORAT PENDIDIKAN TINGGI VOKASI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website: www.polisriwijaya.ac.id E-mail: info@polsri.ac.id

PELAKSANAAN REVISI LAPORAN AKHIR

Mahasiswa berikut,

Nama	:	Natasya Ramadhani
NIM	:	061930322852
Jurusan/Program Studi	:	Teknik Elektro/DIII Teknik Elektronika
Judul Laporan Akhir	:	Rancang Bangun Simulator Kebakaran pada Mesin Pesawat DC-9 Berbasis *Internet of Things* (IoT)

Telah melaksanakan revisi terhadap Laporan Akhir yang diujikan pada hari Kamis tanggal 28 bulan Juli tahun 2022. Pelaksanaan revisi terhadap Laporan Akhir tersebut telah disetujui oleh Dosen Penguji yang memberikan revisi:

No.	Komentar	Nama Dosen Penguji *)	Tanggal	Tanda Tangan
1.	Telah direvisi	Evelina, S.T., M.Kom. NIP. 196411131989032001	22/8/22	
2.	sudah direvisi	Ekawati Prihatini, S.T., M.T. NIP. 197903102002122005	22/8/22	
3.	telah direvisi	Anton Firmansyah, S.T., M.T. NIP. 197509242008121001	10/8/22	

Palembang, 22/8/22

Ketua Penguji **),

(Evelina, S.T., M.Kom.)
NIP. 196411131989032001

Catatan:
*) Dosen penguji yang memberikan revisi saat ujian laporan akhir.
**) Dosen penguji yang ditugaskan sebagai Ketua Penguji saat ujian LA.
Lembaran pelaksanaan revisi ini harus dilampirkan dalam Laporan Akhir.

LAMPIRAN G

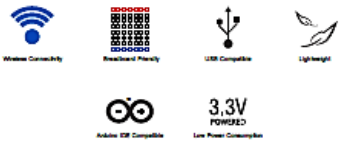
1. NodeMCU ESP8266

NodeMCU ESP8266 ESP-12E WiFi Development Board

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the DevKit. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.

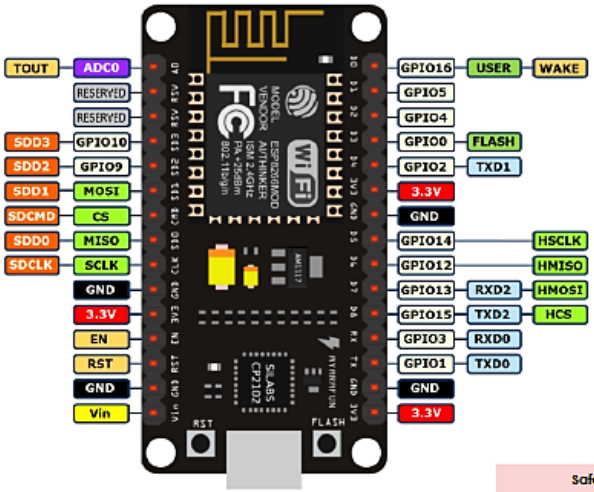
Features

- Version : DevKit v1.0
- Breadboard Friendly
- Light Weight and small size.
- 3.3V operated, can be USB powered.
- Uses wireless protocol 802.11b/g/n.
- Built-in wireless connectivity capabilities.
- Built-in PCB antenna on the ESP-12E chip.
- Capable of PWM, I2C, SPI, UART, 1-wire, 1 analog pin.
- Uses CP2102 USB Serial Communication interface module.
- Arduino IDE compatible (extension board manager required).
- Supports Lua (alike node.js) and Arduino C programming language.



PINOUT DIAGRAM

NodeMCU ESP8266 v1.0



Safety Precaution
All GPIO runs at 3.3V !!

Source
https://iotbytes.wordpress.com/nodemcu-pinout/

Front View

Front View

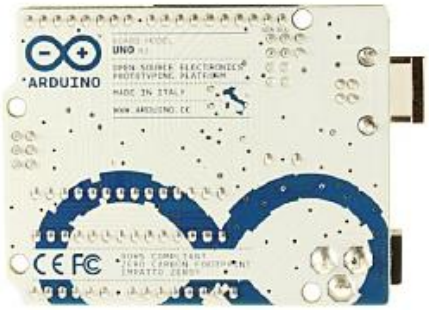
Specifications of ESP-12E WiFi Module

Wireless Standard	IEEE 802.11 b/g/n
Frequency Range	2.412 - 2.484 GHz
Power Transmission	802.11b : +16 ± 2 dBm (at 11 Mbps) 802.11g : +14 ± 2 dBm (at 54 Mbps) 802.11n : +13 ± 2 dBM (at HT20, MCS7)
Receiving Sensitivity	802.11b : -93 dBm (at 11 Mbps, CCK) 802.11g : -85 dBm (at 54 Mbps, OFDM) 802.11n : -82 dBm (at HT20, MCS7)
Wireless Form	On-board PCB Antenna
IO Capability	UART, I2C, PWM, GPIO, 1 ADC
Electrical Characteristic	3.3 V Operated 15 mA output current per GPIO pin 12 - 200 mA working current Less than 200 uA standby current
Operating Temperature	-40 to +125 °C
Serial Transmission	110 - 921600 bps, TCP Client 5
Wireless Network Type	STA / AP / STA + AP
Security Type	WEP / WPA-PSK / WPA2-PSK
Encryption Type	WEP64 / WEP128 / TKIP / AES
Firmware Upgrade	Local Serial Port, OTA Remote Upgrade
Network Protocol	IPv4, TCP / UDP / FTP / HTTP
User Configuration	AT + Order Set, Web Android / iOS, Smart Link APP

2. Arduino

Arduino Uno

Arduino Uno R3 Front



Arduino Uno R3 Back

Arduino Uno R2 Front

Arduino Uno SMD

Arduino Uno Front

Arduino Uno Back

Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.
The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

> Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.
> Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V

Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer)
Schematic: arduino-uno-Rev3-schematic.pdf
Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.
External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.
The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.
The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.**This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).
A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.
The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.
The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).
You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.
The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

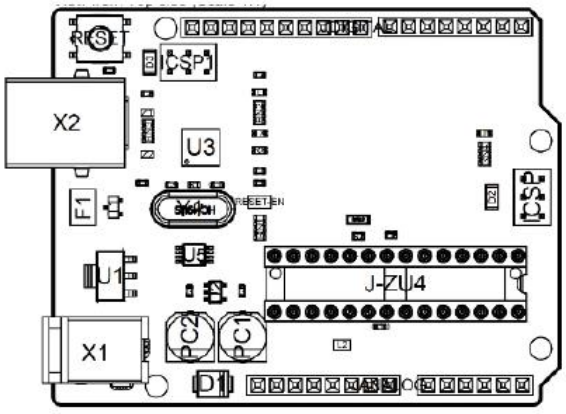
Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

3 Functional Overview

3.1 Board Topology

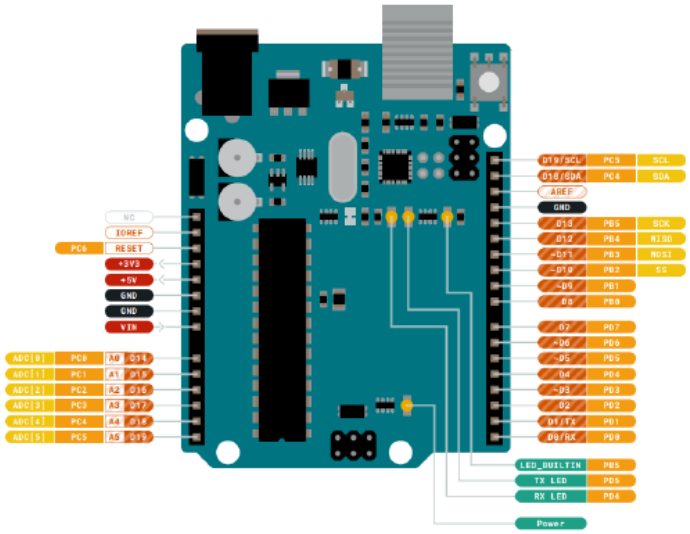
Top view



Board topology

Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through hole 6)
J-ZU4	ATMEGA328P Module	ICSP1	Pin header connector (through hole 6)
Y1	ECS-160-20-4X-DU Oscillator		

5 Connector Pinouts



Pinout

5.1 JANALOG

Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

5.2 JDIGITAL

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

3. MAX6675

EVALUATION KIT AVAILABLE

Click here to ask about the production status of specific part numbers.

MAX6675

Cold-Junction-Compensated K-Thermocouple-to-Digital Converter (0°C to +1024°C)

General Description

The MAX6675 performs cold-junction compensation and digitizes the signal from a type-K thermocouple. The data is output in a 12-bit resolution, SPI-compatible, read-only format.

This converter resolves temperatures to 0.25°C, allows readings as high as +1024°C, and exhibits thermocouple accuracy of 8 LSBs for temperatures ranging from 0°C to +700°C.

The MAX6675 is available in a small, 8-pin SO package.

Applications

- Industrial
- Appliances
- HVAC

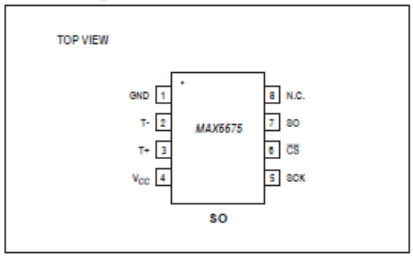
Features

- Direct Digital Conversion of Type -K Thermocouple Output
- Cold-Junction Compensation
- Simple SPI-Compatible Serial Interface
- 12-Bit, 0.25°C Resolution
- Open Thermocouple Detection

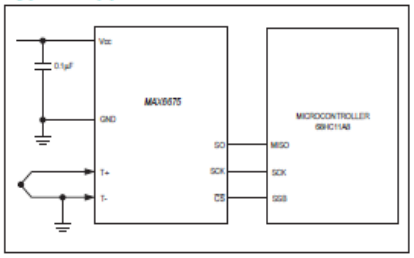
Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX6675ISA	-20°C to +85°C	8 SO

Pin Configuration



Typical Application Circuit



19-2235; Rev 3; 6/21

maxim integrated.

Absolute Maximum Ratings

Supply Voltage (V_{CC} to GND) ..... -0.3V to +6V
SO, SCK, $\overline{CS}$, T-, T+ to GND .....-0.3V to V_{CC} + 0.3V
SO Current 50mA
ESD Protection (Human Body Model) ..... ±2000V
Continuous Power Dissipation (T_A = +70°C)
8-Pin SO (derate 5.88mW/°C above +70°C) 471mW
Operating Temperature Range ..... -20°C to +85°C
Storage Temperature Range ..... -65°C to +150°C
Junction Temperature ..... +150°C
SO Package
Vapor Phase (60s) .....+215°C
Infrared (15s) .....+220°C
Lead Temperature (soldering, 10s) ..... +300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Electrical Characteristics

(V_{CC} = +3.0V to +5.5V, T_A = -20°C to +85°C, unless otherwise noted. Typical values specified at +25°C.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Temperature Error		$T_{THERMOCOUPLE}$ = +700°C, T_A = +25°C (Note 2)	V_{CC} = +3.3V	-5		+5	LSB
			V_{CC} = +5V	-6		+6	
		$T_{THERMOCOUPLE}$ = 0°C to +700°C, T_A = +25°C (Note 2)	V_{CC} = +3.3V	-8		+8	
			V_{CC} = +5V	-9		+9	
		$T_{THERMOCOUPLE}$ = +700°C to +1000°C, T_A = +25°C (Note 2)	V_{CC} = +3.3V	-17		+17	
			V_{CC} = +5V	-19		+19	
Thermocouple Conversion Constant					10.25		µV/LSB
Cold-Junction Compensation Error		T_A = -20°C to +85°C (Note 2)	V_{CC} = +3.3V	-3.0		+3.0	°C
			V_{CC} = +5V	-3.0		+3.0	
Resolution					0.25		°C
Thermocouple Input Impedance					60		kΩ
Supply Voltage	V_{CC}			3.0		5.5	V
Supply Current	I_{CC}				0.7	1.5	mA
Power-On Reset Threshold		V_{CC} rising		1	2	2.5	V
Power-On Reset Hysteresis					50		mV
Conversion Time		(Note 2)			0.17	0.22	s
SERIAL INTERFACE							
Input Low Voltage	V_{IL}					0.3 x V_{CC}	V
Input High Voltage	V_{IH}			0.7 x V_{CC}			V
Input Leakage Current	I_{LEAK}	V_{IN} = GND or V_{CC}				±5	µA
Input Capacitance	C_{IN}				5		pF

Electrical Characteristics (continued)

(V_{CC} = +3.0V to +5.5V, T_A = -20°C to +85°C, unless otherwise noted. Typical values specified at +25°C.) (Note 1)

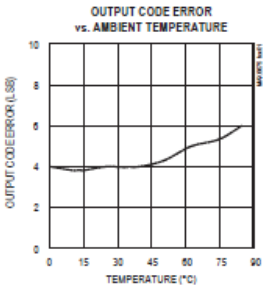
PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output High Voltage	V_{OH}	I_{SOURCE} = 1.6mA	V_{CC} - 0.4			V
Output Low Voltage	V_{OL}	I_{SINK} = 1.6mA			0.4	V
TIMING						
Serial Clock Frequency	f_{SCL}				4.3	MHz
SCK Pulse High Width	t_{CH}		100			ns
SCK Pulse Low Width	t_{CL}		100			ns
CSB Fall to SCK Rise	t_{CSS}	C_L = 10pF	100			ns
CSB Fall to Output Enable	t_{DV}	C_L = 10pF			100	ns
CSB Rise to Output Disable	t_{TR}	C_L = 10pF			100	ns
SCK Fall to Output Data Valid	t_{DO}	C_L = 10pF			100	ns

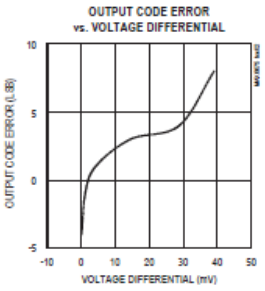
Note 1: All specifications are 100% tested at T_A = +25°C. Specification limits over temperature (T_A = T_{MIN} to T_{MAX}) are guaranteed by design and characterization, not production tested.

Note 2: Guaranteed by design. Not production tested.

Typical Operating Characteristics

(V_{CC} = +3.3V, T_A = +25°C, unless otherwise noted.)





Pin Description

PIN	NAME	FUNCTION
1	GND	Ground
2	T-	Alumel Lead of Type-K Thermocouple. Should be connected to ground externally.
3	T+	Chromel Lead of Type-K Thermocouple
4	V_{CC}	Positive Supply. Bypass with a 0.1µF capacitor to GND.
5	SCK	Serial Clock Input
6	$\overline{CS}$	Chip Select. Set $\overline{CS}$ low to enable the serial interface.
7	SO	Serial Data Output
8	N.C.	No Connection

Detailed Description

The MAX6675 is a sophisticated thermocouple-to-digital converter with a built-in 12-bit analog-to-digital converter (ADC). The MAX6675 also contains cold-junction compensation sensing and correction, a digital controller, an SPI-compatible interface, and associated control logic.

The MAX6675 is designed to work in conjunction with an external microcontroller (µC) or other intelligence in thermostatic, process-control, or monitoring applications.

Temperature Conversion

The MAX6675 includes signal-conditioning hardware to convert the thermocouple's signal into a voltage compatible with the input channels of the ADC. The T+and T- inputs connect to internal circuitry that reduces the introduction of noise errors from the thermocouple wires.

Before converting the thermoelectric voltages into equivalent temperature values, it is necessary to compensate for the difference between the thermocouple cold-junction side (MAX6675 ambient temperature) and a 0°C virtual reference. For a type-K thermocouple, the voltage changes by 41µV/°C, which approximates the thermocouple characteristic with the following linear equation:

$$V_{OUT} = (41\mu V / °C) \times (T_R - T_{AMB})$$

Where:

V_{OUT} is the thermocouple output voltage (µV).

T_R is the temperature of the remote thermocouple junction (°C).

T_{AMB} is the ambient temperature (°C).

Cold-Junction Compensation

The function of the thermocouple is to sense a difference in temperature between two ends of the thermocouple wires. The thermocouple's hot junction can be read from 0°C to +1023.75°C. The cold end (ambient temperature of the board on which the MAX6675 is mounted) can only range from -20°C to +85°C. While the temperature at the cold end fluctuates, the MAX6675 continues to accurately sense the temperature difference at the opposite end.

The MAX6675 senses and corrects for the changes in the ambient temperature with cold-junction compensation. The device converts the ambient temperature reading into a voltage using a temperature-sensing diode. To make the actual thermocouple temperature measurement, the MAX6675 measures the voltage from the thermocouple's output and from the sensing diode. The device's internal circuitry passes the diode's voltage (sensing ambient temperature) and thermocouple voltage (sensing remote temperature minus ambient temperature) to the conversion function stored in the ADC to calculate the thermocouple's hot-junction temperature.

Optimal performance from the MAX6675 is achieved when the thermocouple cold junction and the MAX6675 are at the same temperature. Avoid placing heat-generating devices or components near the MAX6675 because this may produce cold-junction-related errors.

Digitization

The ADC adds the cold-junction diode measurement with the amplified thermocouple voltage and reads out the 12-bit result onto the SO pin. A sequence of all zeros means the thermocouple reading is 0°C. A sequence of all ones means the thermocouple reading is +1023.75°C.

Applications Information

Serial Interface

The *Typical Application Circuit* shows the MAX6675 interfaced with a microcontroller. In this example, the MAX6675 processes the reading from the thermocouple and transmits the data through a serial interface. Force $\overline{CS}$ low and apply a clock signal at SCK to read the results at SO. Forcing $\overline{CS}$ low immediately stops any conversion process. Initiate a new conversion process by forcing $\overline{CS}$ high.

Force $\overline{CS}$ low to output the first bit on the SO pin. A complete serial interface read requires 16 clock cycles. Read the 16 output bits on the falling edge of the clock. The first bit, D15, is a dummy sign bit and is always zero. Bits D14–D3 contain the converted temperature in the order of MSB to LSB. Bit D2 is normally low and goes high when the thermocouple input is open. D1 is low to provide a device ID for the MAX6675 and bit D0 is three-state.

Figure 1a is the serial interface protocol and Figure 1b shows the serial interface timing. Figure 2 is the SO output.

Open Thermocouple

Bit D2 is normally low and goes high if the thermocouple input is open. In order to allow the operation of the open thermocouple detector, T- must be grounded. Make the ground connection as close to the GND pin as possible.

Noise Considerations

The accuracy of the MAX6675 is susceptible to power-supply coupled noise. The effects of power-supply noise can be minimized by placing a 0.1µF ceramic bypass capacitor close to the supply pin of the device.

Thermal Considerations

Self-heating degrades the temperature measurement accuracy of the MAX6675 in some applications. The magnitude of the temperature errors depends on the thermal conductivity of the MAX6675 package, the mounting technique, and the effects of airflow. Use a large ground plane to improve the temperature measurement accuracy of the MAX6675.

The accuracy of a thermocouple system can also be improved by following these precautions:

- Use the largest wire possible that does not shunt heat away from the measurement area.
- If small wire is required, use it only in the region of the measurement and use extension wire for the region with no temperature gradient.
- Avoid mechanical stress and vibration, which could strain the wires.
- When using long thermocouple wires, use a twisted-pair extension wire.
- Avoid steep temperature gradients.
- Try to use the thermocouple wire well within its temperature rating.
- Use the proper sheathing material in hostile environments to protect the thermocouple wire.
- Use extension wire only at low temperatures and only in regions of small gradients.
- Keep an event log and a continuous record of thermocouple resistance.

Reducing Effects of Pick-Up Noise

The input amplifier (A1) is a low-noise amplifier designed to enable high-precision input sensing. Keep the thermocouple and connecting wires away from electrical noise sources.

LAMPIRAN H

1. AMM DC-9 *Chapter* 26 *Fire Protection*

GENERAL - DESCRIPTION AND OPERATION

1. Description

A. Fire protection consists of a detection system and an extinguishing system. The detection system provides the means to detect a fire and alert the crew by aural and visual indications. The extinguishing system provides the means to extinguish a fire.

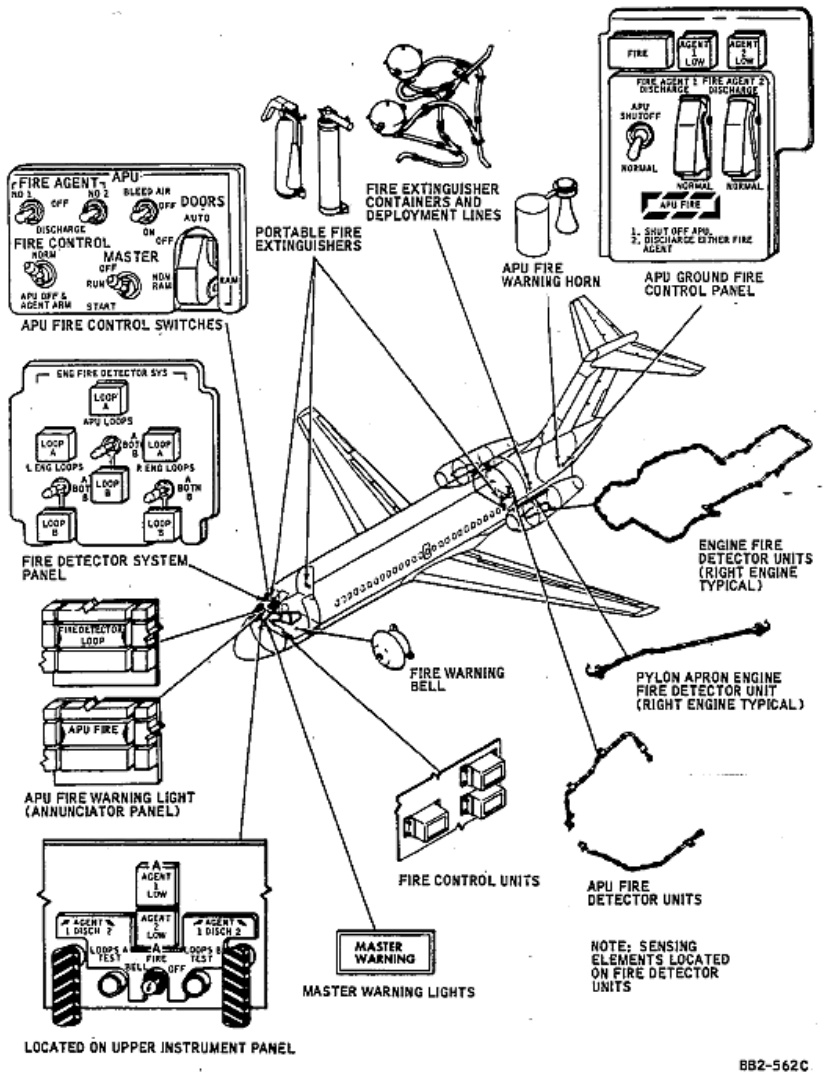
B. Provisions for fire detection are installed in the left and right engine compartments and the auxiliary power unit (APU) compartment. The detection system is basically a dual loop arrangement using two separate fire sensing element loops and a fire detector control unit for each designated fire area. The detection system in each fire area is capable of detecting a fire with one sensing element loop ground faulted or open. Fire warning for the engine areas is indicated by lights located in the left and right engine fire control handles in the flight compartment, and a fire warning bell. Fire warning for the APU compartment is indicated by an APU fire light in the annunciator panel, master warning lights on the glareshield in the flight compartment, the fire warning horn, and a fire light on the APU ground fire control panel. Provisions for monitoring and checking integrity of the detection system are provided by a fire detector system panel, a fire detector loop light, and loop test switches located in the flight compartment.

C. A fixed fire extinguishing system is provided for the engine and APU compartments. The system consists of two fire extinguishing agent containers, providing two-shot extinguishing capability, deployment lines, and fire extinguishing discharge controls for each area. The discharge controls provide a means of selecting either fire extinguishing agent container for discharge. Discharge controls for the engines are a left and right engine fire control handle located on the upper instrument panel. Discharge controls for the APU compartment are two switches located on the APU control panel in the flight compartment, and two switches located on the APU ground fire control panel.

D. Low agent indicating lights, one for each of the fixed system fire extinguisher containers, are located on the upper instrument panel, and on the APU ground fire control panel. When a container is discharged, the corresponding lights come on.

E. Portable fire extinguishers, located in the flight and passenger compartments, provide fire extinguishing capability in these compartments or areas accessible from them.

PRINTED IN U.S.A.



BB2-562C

Fire Protection System
Figure 1

PRINTED IN U.S.A.

2. AMM BOEING 737 NG *Chapter* 26 *Fire Protection*

FIRE PROTECTION - ENGINE FIRE DETECTION - ENGINE FIRE DETECTORS

Purpose

The engine fire detectors monitor for high temperatures in the engine area.

General Description

Each engine has eight detectors. The detectors monitor four sections of the engine. In each section, two detectors attach to a support tube and make an assembly. An assembly has one detector from loop A and one from loop B.

Physical Description

The engine fire detectors have these parts:

- Overheat, fire, and fault pressure switches
- Resistors
- Terminal stud
- A stainless steel, gas charged tube.

Location

These are the locations of the engine fire detectors:

- Two on the upper fan case section
- Two on the lower fan case section
- Two on the left core section
- Two on the right core section.

Detector Characteristics

This table shows the temperature characteristics of the detectors.

Detector Location	Overheat	Fire
Upper Fan Case	345F (174C)	580F (304C)
Lower Fan Case	345F (174C)	580F (304C)
Left Core Section	650F (343C)	850F (454C)
Right Core section	650F (343C)	850F (454C)

Operation

The three pressure switches sense these conditions:

- Overheat
- Fire
- Fault (loss of gas pressure).

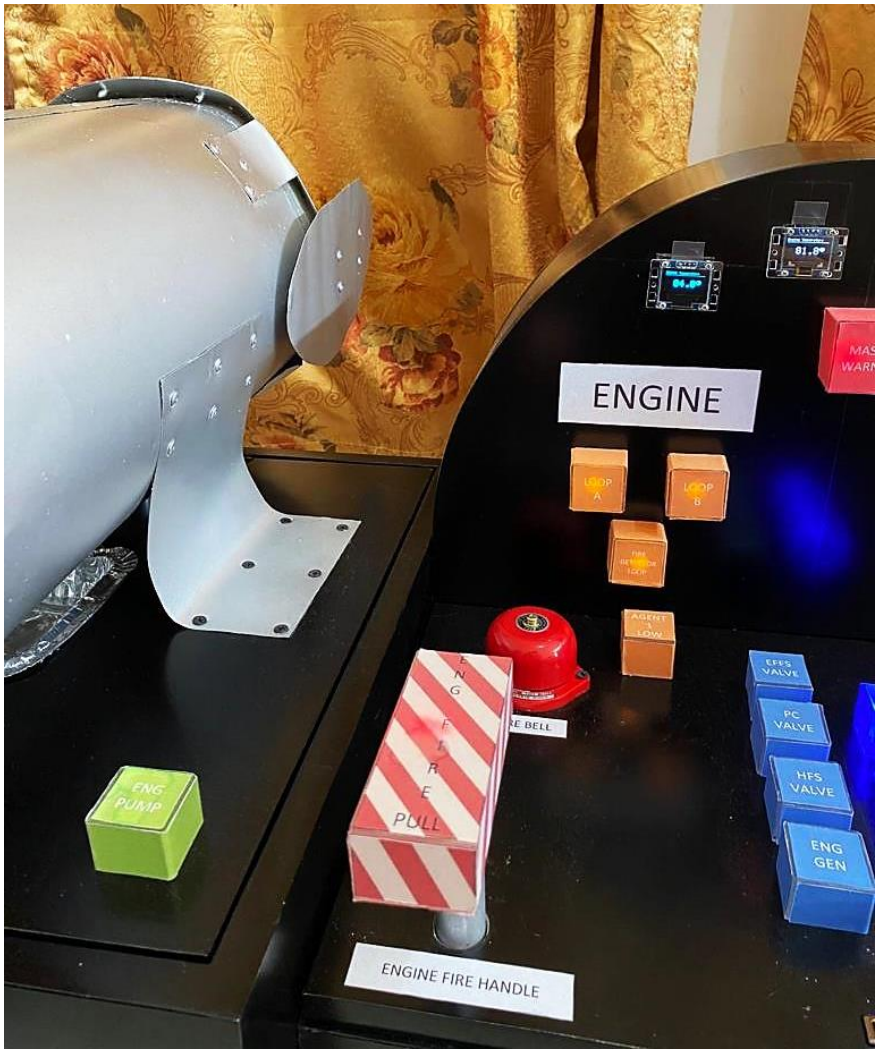
Gas pressure in the sense tube holds the fault pressure switch in the closed position. The other two pressure switches close when the gas pressure increases because of an overheat or fire condition.

The overheat and fire signals go to the engine and the APU fire detection module. This module supplies overheat or fire indication in the flight compartment.

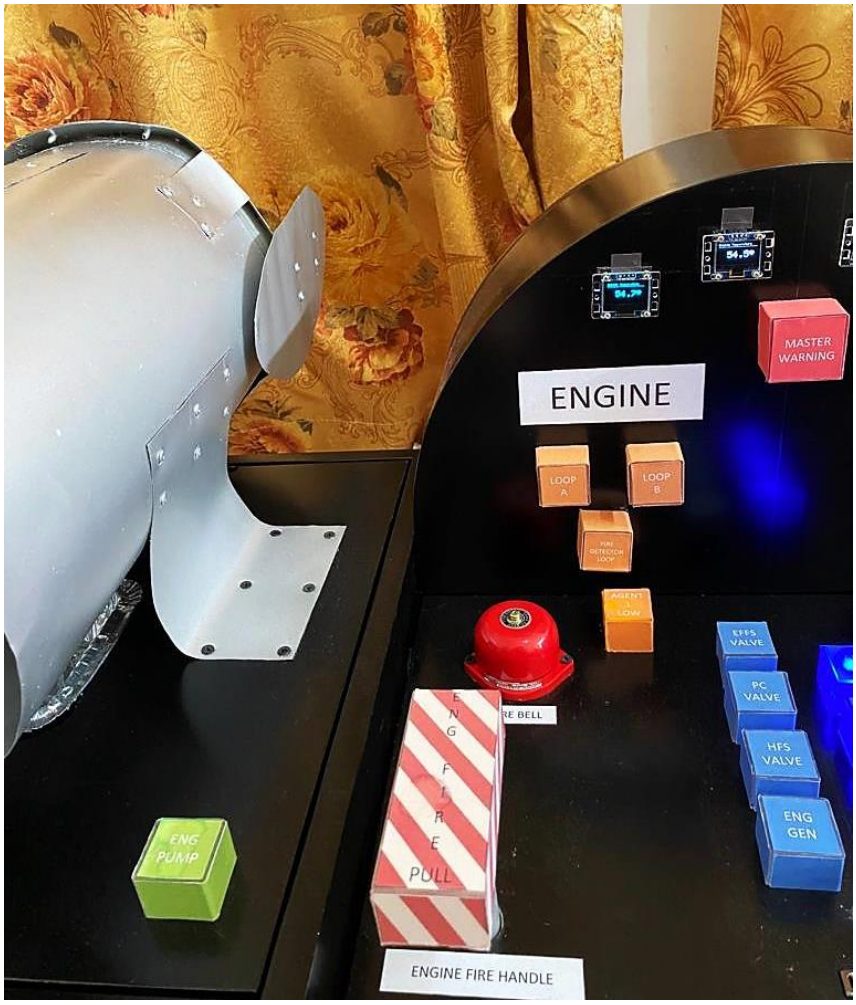
If the pressure in the sense tube decreases, the fault switch opens. This switch sends the fault signal to the engine and APU fire detection module.

LAMPIRAN I

Gambar 1. Kondisi Alat saat Keadaan Normal (<76°C)



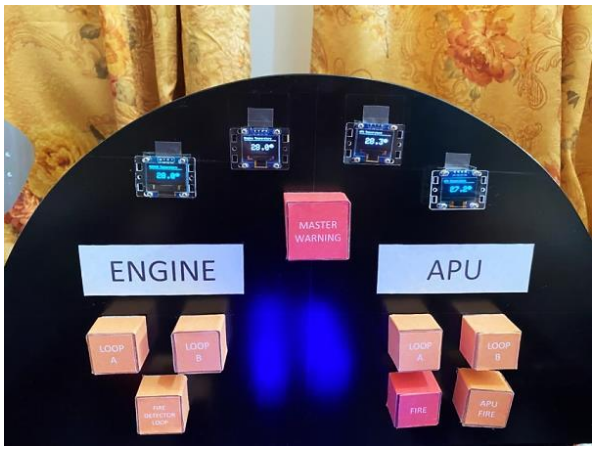
Gambar 2. Kondisi Alat ketika Terdeteksi Kebakaran (≥76°C)



Gambar 3. Kondisi Alat setelah Terjadi Kebakaran (<76°C)

Gambar 4. Pengukuran Tegangan Sensor *Thermocouple*

Gambar 5. *Control Panel* Simulator



Gambar 6. Bagian Atas *Control Panel*

Gambar 7. Bagian Bawah *Control Panel*

Gambar 8. Tampak Depan Miniatur *Engine*

Gambar 9. Tampak Atas Miniatur *Engine*

LAMPIRAN J

STANDARD OPERATING PROCEDURE (SOP)

SIMULATOR *ENGINE FIRE* PADA PESAWAT DC-9

BERBASIS *INTERNET OF THINGS (*IOT*)*

Disusun Oleh:

Natasya Ramadhani

061930322852

1. Proses simulasi dimulai.
2. Sebelum menghidupkan alat simulator, hidupkan jaringan *hotspot* WiFi yang digunakan untuk sistem IoT-nya dan pastikan bahwa koneksinya baik serta hubungkan *smartphone* yang digunakan untuk mengoperasikan aplikasi Blynk ke jaringan WiFi yang sama.
3. Lalu, hubungkan kedua *adaptor* untuk NodeMCU ESP8266 dan Arduino Uno R3 serta steker bel pada sumber listrik AC untuk menghidupkan alat simulatornya.
4. Pastikan bahwa kedua OLED *display* menyala dan menunjukkan suhu ruang pada saat alat dihidupkan, empat LED sistem berwarna biru menyala, dan seluruh indikator lainnya dalam kondisi tidak aktif.
5. Buka aplikasi Blynk pada *smartphone* dan pilih *template device* yang telah dibuat untuk melihat apakah NodeMCU ESP8266 dalam keadaan *online* dan menunjukkan suhu yang terdeteksi atau tidak.
6. Selanjutnya, untuk mensimulasikan keadaan ketika terjadi kebakaran, arahkan dua pemantik api ke dua sensor *thermocouple* yang berada di sebelah kanan untuk sistem deteksi dan sebelah kiri untuk sistem pemadam.
7. Pastikan bahwa suhu yang ditampilkan oleh OLED *display* dan aplikasi Blynk mengalami kenaikan yang stabil.
8. Ketika suhu mencapai ≥76°C, indikator sistem deteksi berupa *engine fire warning lights* yang terdiri dari LED *master warning*, LED *loop* A, LED *loop* B, LED *fire detector loop*, LED *engine fire* yang terdapat di *engine fire*

handle, bel, dan notifikasi terjadinya kebakaran pada aplikasi Blynk pun akan aktif secara bersamaan.

9. Berikutnya, hal yang harus dilakukan untuk memadamkan api sehingga suhu turun menjadi normal kembali adalah menarik *engine fire handle* sembari menekan *switch* 1 ke posisi *ON* sehingga keempat LED sistem berwarna biru menjadi tidak aktif.
10. Lalu, putar *engine fire handle* ke arah kanan sembari menekan *switch* 2 ke posisi *ON* untuk mengaktifkan *water pump* yang ditandai dengan hidupnya LED *engine pump* yang berwarna hijau.
11. Terakhir, setelah *water pump* aktif dan menyemprotkan air langsung ke dua sensor *thermocouple*, maka suhu akan kembali turun mencapai <76°C, seluruh indikator sistem deteksi menjadi normal kembali, *water pump* akan mati dan LED *low agent light* yang berwarna oranye menyala.
12. Proses simulasi selesai dilaksanakan.

LAMPIRAN K

1. *Fire Detection System*

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "max6675.h"

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

#define OLED_RESET     -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

int thermoDO = 12;
int thermoCS = 15;
int thermoCLK = 14;

MAX6675 thermocouple(thermoCLK, thermoCS, thermoDO);
float temp, a;

#define ledmasterwarning 0
#define ledEngine 2
#define relaybell 13
#define BLYNK_PRINT Serial

#define BLYNK_TEMPLATE_ID           "TMPLKQ6Xc5dK"
#define BLYNK_DEVICE_NAME         "Fire Simulator"
#define BLYNK_AUTH_TOKEN
"51qeNpXNnZ5SMDOyzDJK0ZXiu6wHCR83"
```

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "FIRE PROTECTION";
char pass[] = "GMFPOLSRI";

BlynkTimer timer;

void sendSensor()
{
  temp = (thermocouple.readCelsius()  - 4);
  displayTemp(temp);

  Serial.print("C1 = ");
  Serial.println(temp);
  delay(500);
  Blynk.virtualWrite(V1, temp);
  if (temp >= 76) {
    digitalWrite(ledmasterwarning, HIGH);
    digitalWrite(ledEngine, HIGH);
    digitalWrite(relaybell, LOW);
  }
  if (temp <= 76) {
    digitalWrite(ledEngine, LOW);
    digitalWrite(ledmasterwarning, LOW);
    digitalWrite(relaybell, HIGH);
  }
}
```

```
void displayTemp(float temp) {
  display.clearDisplay();
  display.display();

  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println("Engine Temperature");
  display.println("----------------------");
  display.setCursor(28, 27);
  display.setTextSize(3);
  display.print(temp, 1);
  display.print((char)247);
  display.display();
}

void setup()
{
  Serial.begin(9600);
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for
128x64
    Serial.println(F("SSD1306 allocation failed"));
    for (;;); // Don't proceed, loop forever
  }

  display.clearDisplay();
  display.display();
  pinMode(ledmasterwarning, OUTPUT);
  pinMode(ledEngine, OUTPUT);
  pinMode(relaybell, OUTPUT);
    Blynk.begin(auth, ssid, pass);
```

```
}
void loop()
{
   Blynk.run();
   sendSensor();
}
```

LAMPIRAN L

1. *Engine Fire Extinguisher System*

```
#include "max6675.h"

#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

//#define OLED_ 22

//Adafruit_SSD1306 display(OLED_);


#define SCREEN_WIDTH 128 // OLED display width, in pixels

#define SCREEN_HEIGHT 64 // OLED display height, in pixels


#define OLED_RESET     -1 // Reset pin # (or -1 if sharing Arduino reset pin)

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);


int thermoDO = 4;

int thermoCS = 5;

int thermoCLK = 6;

float k1 = 1; //untuk kalibrasi k1= (0.1 - 1.9)

float k2 = 1; //untuk kalibrasi k1= (0.1 - 1.9)

MAX6675 thermocouple(thermoCLK, thermoCS, thermoDO);

#define pb1 7 //8//6

#define pb2 8

#define led4 12

#define ledpump 10
```

```
#define ledlowagent 11
#define relay 9
#define switch1 digitalRead (pb1)
#define switch2 digitalRead (pb2)

float temp, a, b;

void setup() {
  Serial.begin(9600);
  inisial_oled();
}
void loop() {
  //  thermocouple1();
  algo();
}
```

2. Algo

```
void algo() {
  thermocouple1();
  if (switch1 == 0) {
    digitalWrite(led4, LOW);
    digitalWrite(ledpump, LOW);
    digitalWrite(ledlowagent, LOW);
    digitalWrite(relay, HIGH);
  }
```

```
  if (a >= 76) {
    if (switch1 == 0 && switch2 == 0) {
      digitalWrite(led4, LOW);
      digitalWrite(ledpump, HIGH);
      digitalWrite(ledlowagent, LOW);
      digitalWrite(relay, LOW);
    }
  }
  if (a <= 76) {
    if (switch1 == 0 && switch2 == 0) {
      digitalWrite(ledlowagent, HIGH);
      digitalWrite(relay, HIGH);
      digitalWrite(ledpump, LOW);
      digitalWrite(led4, LOW);
    }
  }
  if (switch1 == 1 && switch2 == 1) {
    digitalWrite(led4, HIGH);
    digitalWrite(ledpump, LOW);
    digitalWrite(ledlowagent, LOW);
    digitalWrite(relay, HIGH);
  }
}
```

3. Inisial Oled

```
void inisial_oled() {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.setCursor(5, 0);
  display.print("ARDUINO");
  display.display();
  pinMode(pb1, INPUT_PULLUP);
  pinMode(pb2, INPUT_PULLUP);
  pinMode(relay, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(ledlowagent, OUTPUT);
  pinMode(ledpump, OUTPUT);
}
void displayTemp(float temp) {
  display.clearDisplay();
  display.display();

  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println("ENGINE Temperature");
  display.println("----------------------");
```

```
  display.setCursor(28, 27);
  display.setTextSize(3);
  display.print(temp, 1);
  display.print((char)247);
  display.display();
}
```

4. Inisial Sensor

```
void thermocouple1() {
  a = (thermocouple.readCelsius()+1);
    displayTemp(a);
  Serial.print("C = ");
  Serial.println(a);

  delay(300);
}
```