

# TPA81 Thermopile Array

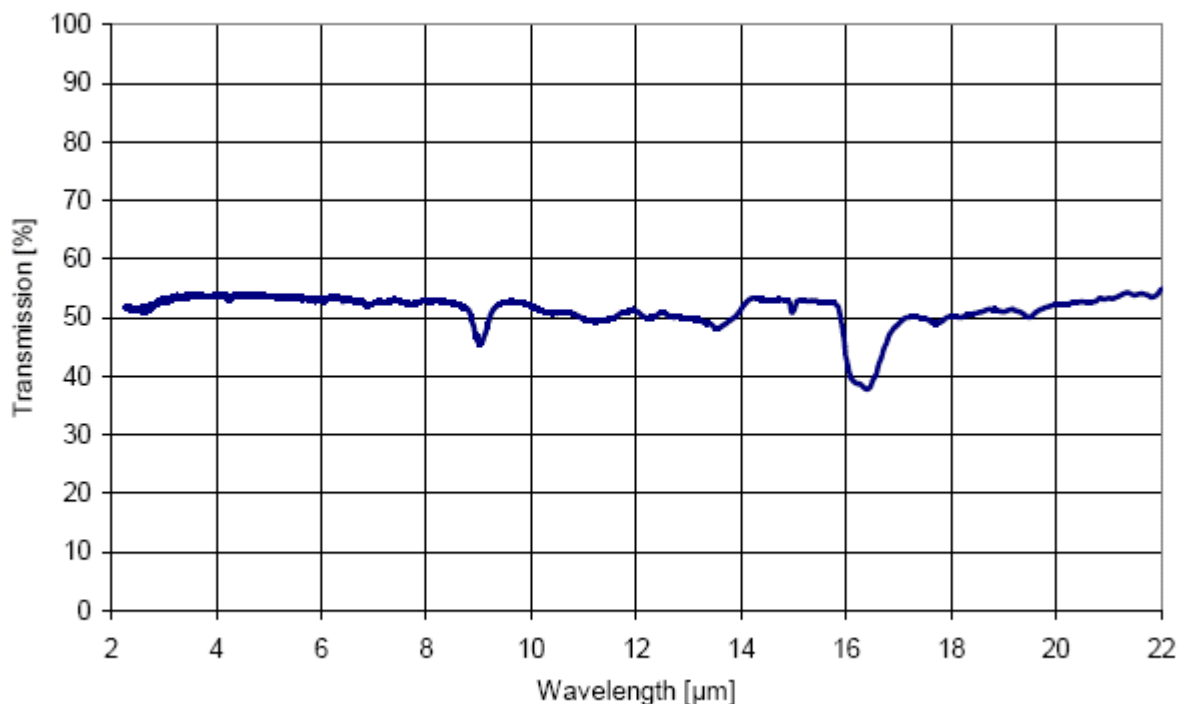
## Technical Specification

### Introduction

The TPA81 is a thermopile array detecting infra-red in the 2 $\mu$ m-22 $\mu$ m range. This is the wavelength of radiant heat. The Pyro-electric sensors that are used commonly in burglar alarms and to switch on outside lights, detect infra-red in the same waveband. These Pyro-electric sensors can only detect a change in heat levels though - hence they are movement detectors. Although useful in robotics, their applications are limited as they are unable to detect and measure the temperature of a static heat source. Another type of sensor is the thermopile array. These are used in non-contact infra-red thermometers. They have a very wide detection angle or field of view (FOV) of around 100° and need either shrouding or a lens or commonly both to get a more useful FOV of around 12°. Some have a built in lens. More recently sensors with an array of thermopiles, built in electronics and a silicon lens have become available. This is the type used in the TPA81. It has a array of eight thermopiles arranged in a row. The TPA81 can measure the temperature of 8 adjacent points simultaneously. The TPA81 can also control a servo to pan the module and build up a thermal image. The TPA81 can detect a candle flame at a range 2 metres (6ft) and is unaffected by ambient light!

### Spectral Response

The response of the TPA81 is typically 2 $\mu$ m to 22 $\mu$ m and is shown below:



### Field of View (FOV)

The typical field of view of the TPA81 is 41° by 6° making each of the eight pixels 5.12° by 6°. The array of eight pixels is orientated along the length of the PCB - that's from top to bottom in the diagram below. Pixel number one is nearest the tab on the sensor - or at the bottom in the diagram below.

### Sensitivity

Here's some numbers from one of our test modules:

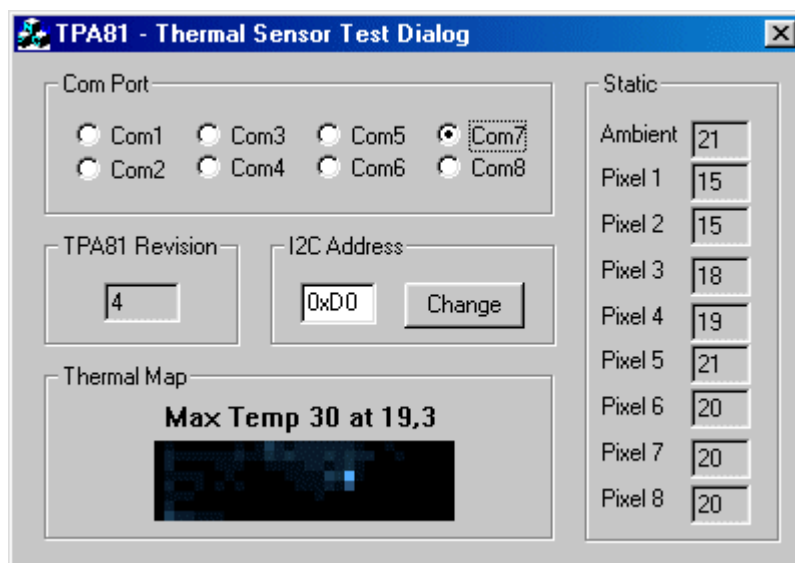
For a candle, the numbers for each of the eight pixels at a range of 1 meter in a cool room at 12°C are:

11 10 11 12 12 29 15 13 (All ȳ)

You can see the candle showing up as the 29ȳ reading. At a range of 2 meters this reduces to 20ȳ - still around 8ȳ above ambient and easily detectable. At 0.6 meter (2ft) its around 64ȳ. At 0.3 meter (1ft) its 100ȳ+.

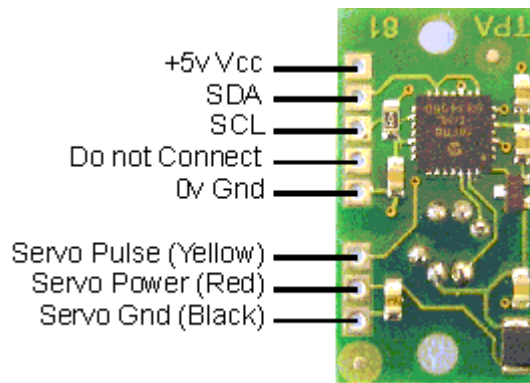
In a warmer room at 18ȳ, the candle measures 27ȳ at 2 meters. This is because the candle only occupies a small part of the sensors field of view and the candles point heat source is added to the back ground ambient - not swamped by it. A human at 2 meters will show up as around 29ȳ with a background 20ȳ ambient.

The following is a snapshot of our test program. It displays a 32x8 bitmap produced by using a servo to pan the sensor. If you want a copy of this windows based program, [its here](#), but you will need an RF04/CM02 to connect the TPA81 to your PC. Here you can see a candle flame about a meter away showing up as the bright spot.



## Connections

All communication with the TPA81 is via the I2C bus. If you are unfamiliar with the I2C bus, there is a [tutorial](#) which will help. The TPA81 uses our standard I2C 5 pin connection layout. The "Do Not Connect" pin should be left unconnected. It is actually the CPU MCLR line and is used once only in our workshop to program the PIC16F88 on-board after assembly, and has an internal pull-up resistor. The SCL and SDA lines should each have a pull-up resistor to +5v somewhere on the I2C bus. You only need one pair of resistors, not a pair for every module. They are normally located with the bus master rather than the slaves. The TPA81 is always a slave - never a bus master. If you need them, I recommend 1.8k resistors. Some modules such as the OOPic already have pull-up resistors and you do not need to add any more. A servo port will connect directly to a standard RC servo and is powered from the modules 5v supply. We use an HS311. Commands can be sent to the TPA81 to position the servo, the servo pulses are generated by the TPA81 module.



## Registers

The TPA81 appears as a set of 10 registers.

Register	Read	Write
0	Software Revision	Command Register
1	Ambient Temperature (°C)	Used for Calibration - do not write
2	Pixel 1 Temperature (°C)	Used for Calibration - do not write
3	Pixel 2	Used for Calibration - do not write
4	Pixel 3	N/A
5	Pixel 4	N/A
6	Pixel 5	N/A
7	Pixel 6	N/A
8	Pixel 7	N/A
9	Pixel 8	N/A

Only registers 0, 1, 2 and 3 can be written to. Register 0 is the command register and is used to set the servo position and also when changing the TPA81's I2C address. It cannot be read. Reading from register 0 returns the TPA81 software revision. Registers 1, 2 and 3, are used for calibration of the sensor. Do not write to these registers or the sensors calibration data may be erased. (There is protection against this, in that a specific 3 byte command sequence similar to the I2C address change sequence, has to be provided to enable calibration mode). Calibration requires the use of two temperature controlled black body heat sources, unless you have these, you will not be able to calibrate the module. All modules are calibrated in our workshop as part of the testing process.

There are 9 temperature readings available, all in degrees centigrade (°C). Register 1 is the ambient temperature as measured within the sensor. Registers 2-9 are the 8 pixel temperatures. Temperature acquisition is continuously performed and the readings will be correct approx 40mS after the sensor points to a new position.

## Servo Position

Commands 0 to 31 set the servo position. There are 32 steps (0-31) which typically represent 180° rotation on a Hitec HS311 servo. The calculation is  $SERVO\_POS * 60 + 540\mu S$ . So the range of the servo pulse is 0.54mS to 2.4mS in 60µS steps. Writing any other value to the command register will stop the servo pulses.

Command		Action
Decimal	Hex	
0	0x00	Set servo position to minimum
nn	nn	Set servo position
31	0x1F	Set servo position to maximum
160	0xA0	1st in sequence to change I2C address
165	0xA5	3rd in sequence to change I2C address
170	0xAA	2nd in sequence to change I2C address

### Changing the I2C Bus Address

To change the I2C address of the TPA81 you must have only one module on the bus. Write the 3 sequence commands in the correct order followed by the address. Example; to change the address of a TPA81 currently at 0xD0 (the default shipped address) to 0xD2, write the following to address 0xD0; (0xA0, 0xAA, 0xA5, 0xD2 ). These commands must be sent in the correct sequence to change the I2C address, additionally, No other command may be issued in the middle of the sequence. The sequence must be sent to the command register at location 0, which means 4 separate write transactions on the I2C bus. Additionally, there MUST be a delay of at least 50mS between the writing of each byte of the address change sequence. When done, you should label the sensor with its address, if you lose track of the module addresses, the only way to find out what it is to search all the addresses one at a time and see which one responds. The TPA81 can be set to any of eight I2C addresses - 0xD0, 0xD2, 0xD4, 0xD6, 0xD8, 0xDA, 0xDC, 0xDE. The factory default shipped address is 0xD0.