

XETAL

Xetal YFC Sensor

---

PRELIMINARY

# Introducing Xetal YFC



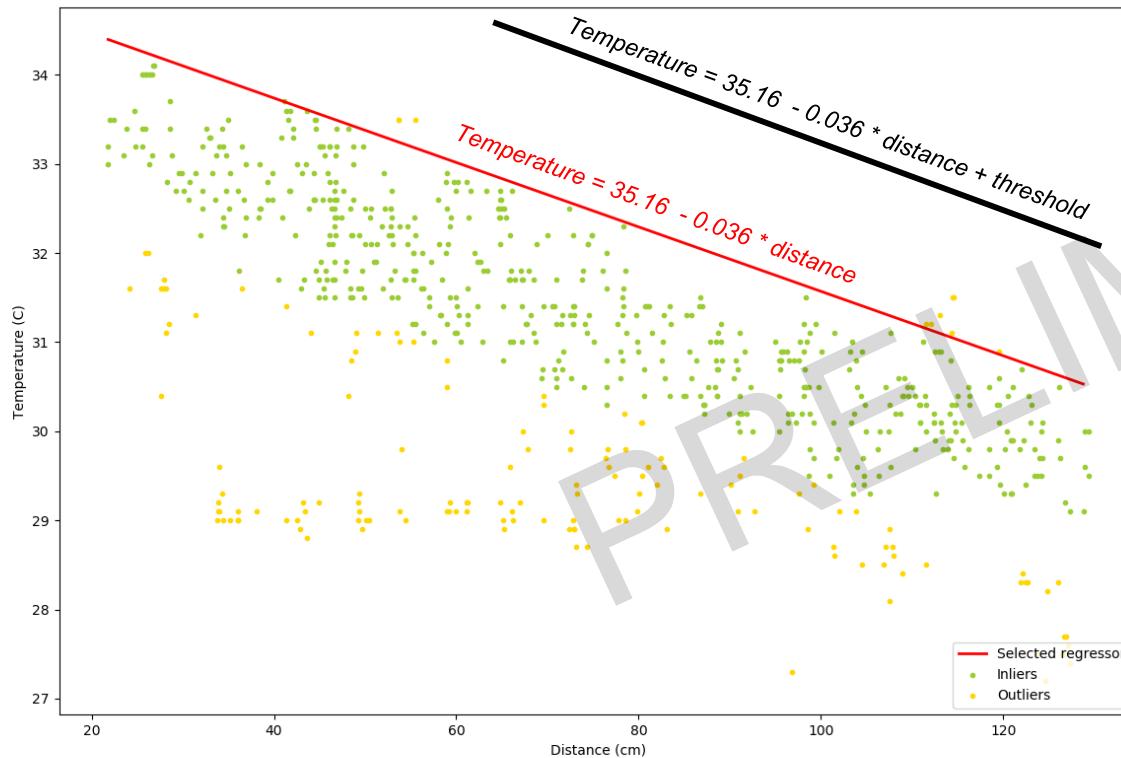
The Xetal YFC sensor combines the power of the Xetal Yugen sensors with an external distance sensor.

Our proven Yugen technology is used to detect a person, isolate his/her face and measure its irradiated temperature. The measure is repeatable and has an error of  $\pm 0.3C$  in the suggested range.

The external distance sensor provides the distance of the person.

The combination of the two data points can be used to generate statistical models in order to detect possible anomalies on people irradiated temperature.

# Example of Xetal YFC usage\*



Xetal YFC sensors can be used to detect a temperature anomaly.

Given enough temperature measurements for different people at different distances, it is possible to generate a regression curve to calculate the most probable maximum temperature for a person at a given distance from the sensor.

We can, then, apply a standard threshold (e.g. 1.5C\*\*) to define a line above which the detected temperature is considered an anomaly.

\* The regression curve has been generated (not by Xetal) on a limited number of points and is given only as example

\*\* Minimum value as from literature about human irradiated measurements

# Xetal YFC: specifications



<b>Model</b>	YFC11025
<b>Detection range</b>	25cm to 90cm (suggested)
<b>Compatible Distance sensors</b>	GP2Y0A02YK0F, VL53L0X
<b>Detection data</b>	Distance, Irradiated temperature
<b>Detection algorithm</b>	Proprietary
<b>Interfaces</b>	USB Serial 115200 1N8
<b>Dimensions</b>	34mm x 34mm
<b>Price (ex works)</b>	Samples (<100 units): EUR 150 Volume: request a quote
<b>Others</b>	The temperature detection algorithm uses no visual or audio data in full respect of GDPR and any other privacy norm



# YFC (mod.11025-USB)

## Protocol Description

### Sensor pins:



### Supported distance sensors:

1. Sharp GP2Y0A02YK0F
2. Adafruit VL53L0X module

### Modes of Operations

The following 4 modes are available:

#### *MODE 0:*

The sensor report temperature every delay\*10 ms.

If human mode is disabled the response is:

-> 0xA1 temp(2) crc8

That can be converted into a float by calculating (temp[0]<<8 | temp[1])/10.

If human mode is enabled, the response is

-> "T=temp \n"

Where temp is a string representing a float.

#### *MODE 1:*

The sensor report temperature and distance when at least for avg/rate seconds the distance sensor report a presence at a distance with an error less than 10cm. Values are reported only if stance is less than 120cm.

If human mode is disabled, the response alternate two responses: one for temperature and one for distance. The first response is

-> 0xA1 temp(2) crc8

That can be converted into a float by calculating  $(\text{temp}[0] \ll 8 | \text{temp}[1]) / 10$ . The second is

-> 0xA2 dist(2) crc8 /

That can be converted into a float by calculating  $(\text{dist}[0] \ll 8 | \text{dist}[1]) / 10$ .

If human mode is enabled, the response is

-> "D=distance T=temp \n"

Where distance and temp are strings representing a float.

#### *MODE 2:*

The sensor report the temperature only when requested using 0x17 command.

If human mode is disabled the response is:

-> 0xA1 temp(2) crc8

That can be converted into a float by calculating  $(\text{temp}[0] \ll 8 | \text{temp}[1]) / 10$ .

If human mode is enabled, the response is

-> "T=temp \n"

Where temp is a string representing a float.

#### *MODE 3 :*

The sensor report temperature and distance when at least for (MODE3\_discard + MODE3\_window)\* avg/rate seconds the distance sensor reports a distance with an error less than 10cm . Values are reported only if stance is less than 120cm .

If human mode is disabled the response is a sequence of three responses. The first response is

-> 0xA1 temp(2) crc8

That can be converted into a float by calculating  $(\text{temp}[0] \ll 8 | \text{temp}[1]) / 10$ . The second is

-> 0xA2 dist(2) crc8 /

That can be converted into a float by calculating  $(\text{dist}[0] \ll 8 | \text{dist}[1]) / 10$ . The third is

-> 0xA3 status(1) crc8

Where status is 1 if a new person was detected, 2 otherwise.

If human mode is enabled, the response is

-> "D=distance T=temp S=status \n"

Where distance and temp are strings representing a float and S is 1 or 2 as above defined

## Communication Protocol

The protocol from the sensors towards a client uses a serial protocol on a USB cable at 115200 8N1. The commands are defines as follows, where -> indicates the command towards the sensor and <- the sensor answer. The size of each field (in bytes) is given in brackets after the field name.

### Command description

*Mode selection (default 1).*

-> 0x15 mode(1) crc8  
<- 0x15 crc8

*Specifies Delay in ms\*10 (default 1s)*

-> 0x18 delay(1) crc8  
<- 0x18 crc8

*Sensor Sampling Rate (1 to 60 per sec) (default 60s)*

-> 0x01 rate(1) crc8  
<- 0x01 crc8

*Size of sliding window in seconds for samples average (1 to 60) (default 15)*

-> 0x02 avg(1) crc8  
<- 0x02 crc8

*Factory Reset (restore defaults settings and reboot)*

-> 0xFF crc8  
<- 0xFF crc8

*Read current settings*

-> 0xFD crc8

<- 0xFD rate(1) avg(1) reserved(6) mode(1) delay(1) human\_mode(1) mode3\_window(1)  
mode3\_discard(1) distance\_sensor\_type(1) eeprom\_status(1) led\_status(1) crc8

*Read EEPROM settings (led status is always 0 in eeprom)*

-> 0xFC crc8

<- 0xFC ate(1) avg(1) reserved(6) mode(1) delay(1) human\_mode(1) mode3\_window(1)  
mode3\_discard(1) distance\_sensor\_type(1) eeprom\_status(1) led\_status(1) crc8

*Save current settings in EEPROM*

-> 0xFB crc8

<- 0xFB crc8

*Set Led status (1 led enabled, 0 led disabled)*

-> 0xFA led\_status(1) crc8

<- 0xFA crc8

*Set output data to be given as human-readable strings (1 enabled, 0 disabled)(default 1)*

-> 0x19 human\_mode(1) crc8

<- 0x19 crc8

*Specifies the window size for MODE3 (max 20 including MODE3\_discard) (default 3)*

-> 0x1B size(1) crc8

<- 0x1B crc8

*Specifies the discard size for MODE3 (max 20 including MODE3\_window) (default 2)*

-> 0x1A size(1) crc8

<- 0x1A crc8

*Select used DISTANCE sensor type (0 sharp, 1 tof) (default 0)*

-> 0x1C size(1) crc8

<- 0x1C crc8

*Request a new temperature reading*

-> 0x17 crc8