

أول sensor استخدمناه هو ال gas quality sensor وعملنا له library خاصة به و هدفه هو قياس متوسط عدد جزيئات عدة غازات في وقت واحد مثل alcohol, Benzene, smoke, CO2 وغيره من الغازات ، لذلك فهو له القدرة على الكشف عن التغيرات في عدد الجزيئات وتوحي قراءاته بوجود عدد جزيئات كبيره وغير طبيعية أو قليله وقد تسبب اختناق النبات بسبب عدم وجود عدد جزيئات كافي في مستودع النبات . ال sensor يتكون من مكثف وملف أساسيين، و من مقاومه متغيره تتغير قيمتها بتغير عدد الجزيئات اسمها RS ومقاومه ثابتة تم من خلالها قياس عدد جزيئات الغازات الطبيعي في الهواء واسمها Ro ومقاومة اضافيه اسمها RL تساعد في حساب قيمة Rs

التغيرات الناتجة عن عدد الجزيئات في المستودع يتم التعبير عنها بتغير أسي يعني التغير في الغاز بياخذ معادله على الشكل $y=c.x^p$ ، لذلك يتم التعبير عن التغير على منحنى log-log ، وتمت معايرة هذا ال sensor بعد الاختبار على أرض الواقع وأخذ ال samples من القيم ال analog باستخدام المعادلة

$$PARA * (((1023/\text{analog value measured by the sensor}) - 1) * RLOAD / RZERO)^{(-PARB)}$$

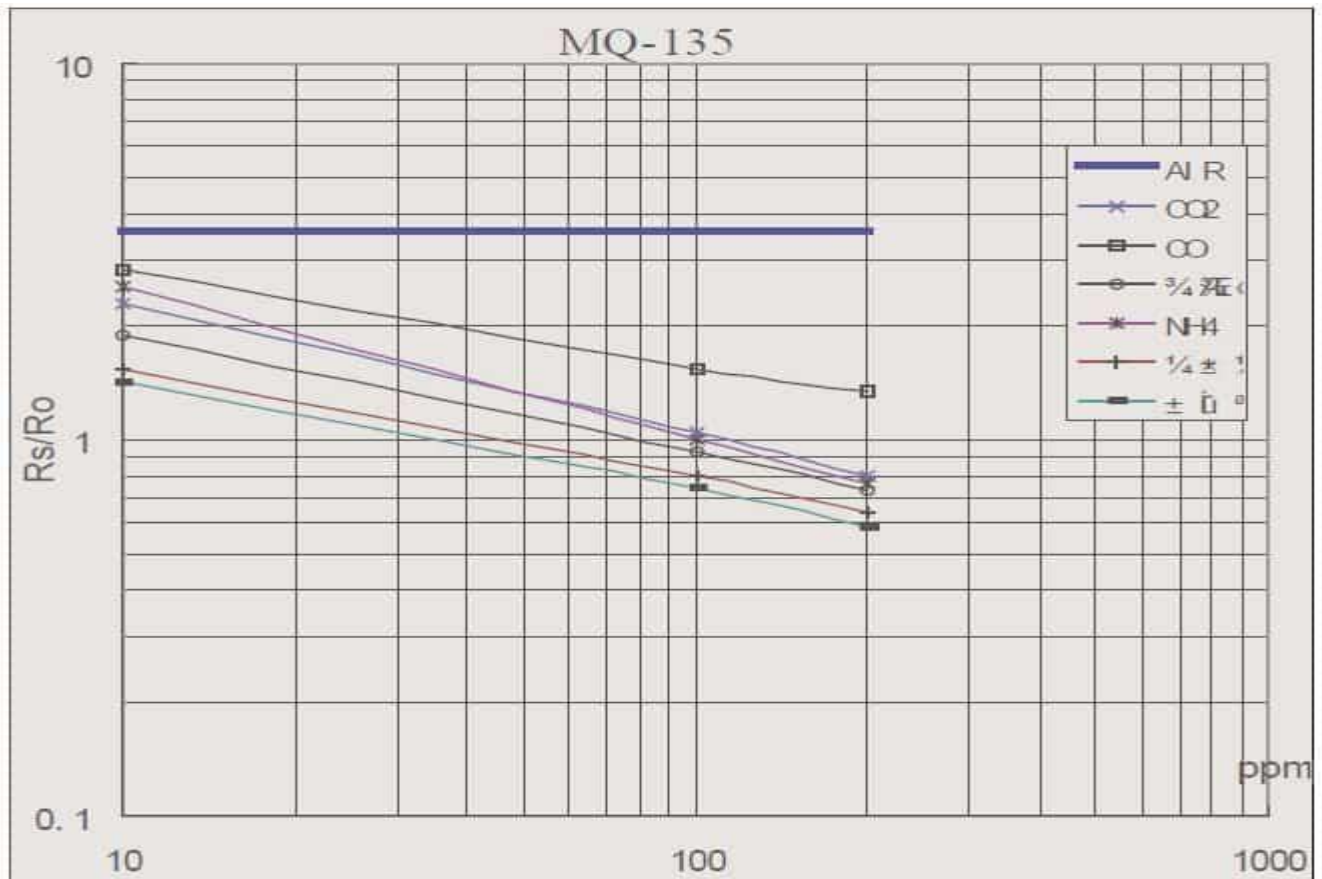
حيث ParaA و ParaB مجرد ثوابت تعبر عن درجة التشتت في المنحنى الاسي

هذه المعادلة مستنتجة من ال data sheet plot ومعتومه على النسبة Rs/Ro

والوحدة التي تعبر عن عدد الجزيئات في الهواء هي parts per million (ppm)

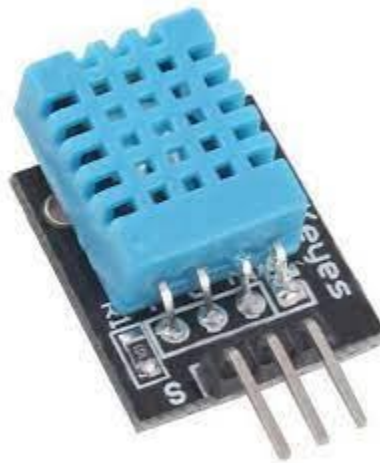
ال sensor مش دقيق اوي لأنه رخيص ولكن يكفيننا معرفة التغير في قيمة ال sensor عند وجود غاز غير مرغوب او عند الكتمة او نقص وجود ال CO2

لازم تأشر على ال curve ده لما تيجي سيرة المعادله اللي بنقيس منها ال gas quality



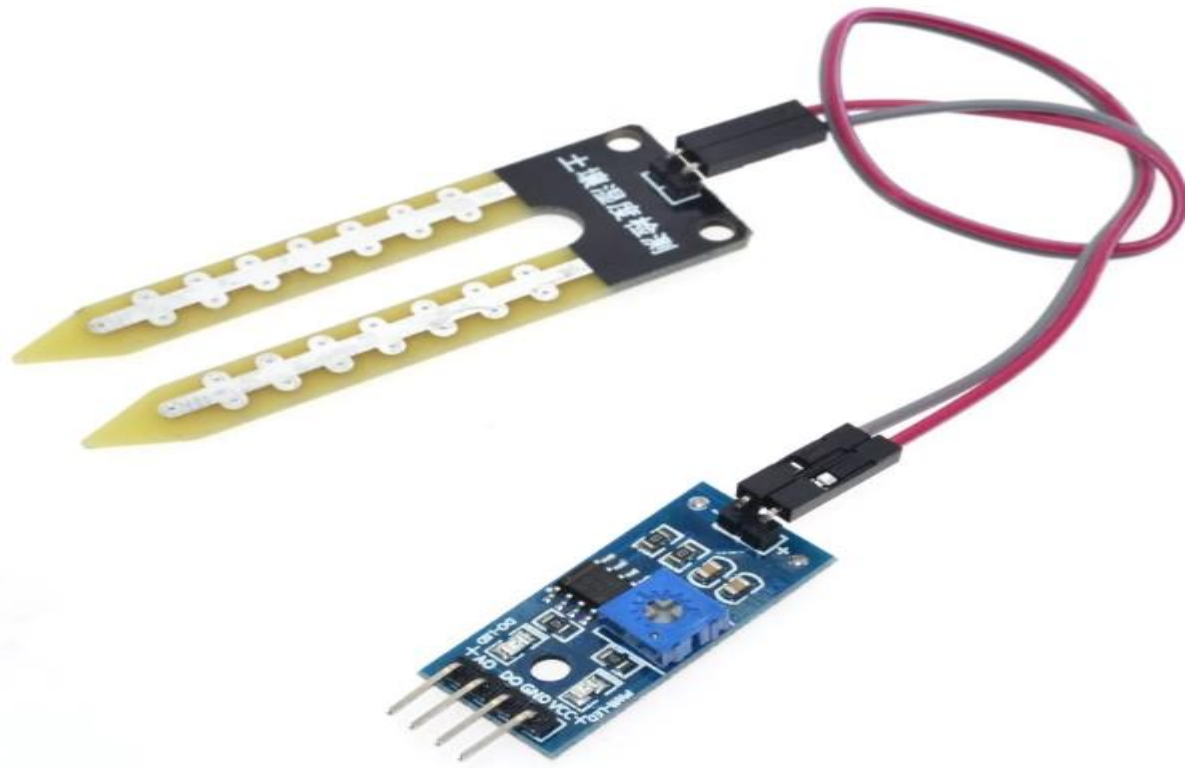


ثاني sensor هو الحرارة والرطوبة وتم استخدام library جاهزة لأنه sensor مشهور



ثالث sensor هو ال resistive moisture sensor يستخدم للتعبير عن نسبة المياه اللي يلتقطها ال sensor عن طريق تغير المقاومة الخاصة بال sensor فكلما كانت نسبة الماء عالية قلت المقاومة وزادت ال conductivity والعكس صحيح وهناك أنواع أخرى تعتمد على استخدام مكثفات و تأخذ قراءات عن طريق تغير قيم ال dielectric constant ولكنها غالية

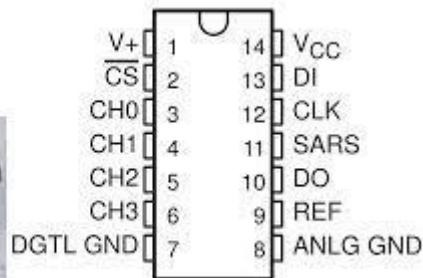
وخرج ال sensor عبارة عن قيم analog لذلك تم استخدام analog to digital converter IC ذات أربعة channels لان ال raspberry pi لا يستطيع استقبال قيم analog الا من خلال I2C AND SPI BUSSES لم نحتاج الى معايرة ال SENSOR لأنه يكفي اختباره حيث انه عندما تنخفض النسبة التي يلتقطها ال SENSOR عن حد معين يتم ضخ الماء المناسب



رابع SENSOR هو ال LIGHT SENSOR حيث انه لابد للضوء كي يقوم النبات بعملية البناء الضوئي وهو حساس ذا مخارج ANALOG فقط وتم استخدام ADC معه مثل ال MOISTURE SENSOR وتم اختباره لمعرفة القيم المناسبة للنبات

فيما يتعلق بالمكتبة الخاصة بال ADC 0834 هذه ال ic بها ٤ channels وال resolution الخاص بها ٨ bits فتتمت معايرتها على النحو التالي :

تم تحديد عدد 16 HIGH AND LOW CLOCKS بحيث كل DURATION تساوي 2 MICRO SECOND و TOTAL DURATION هيساوي $16 * 2 * 10^{-6}$ يعني 32 MICRO SECOND حتى نستطيع أن نأخذ SAMPLES من ال ANALOG VALUES بغض النظر عن فترة ال initialization



of levels (quantization) = $2^8 = 256$
step size = $(5v - 0v) / 256 = 0.0195 v$

5 volts

255

1.5 volts

$1.5 / (5 / 256) = 77$

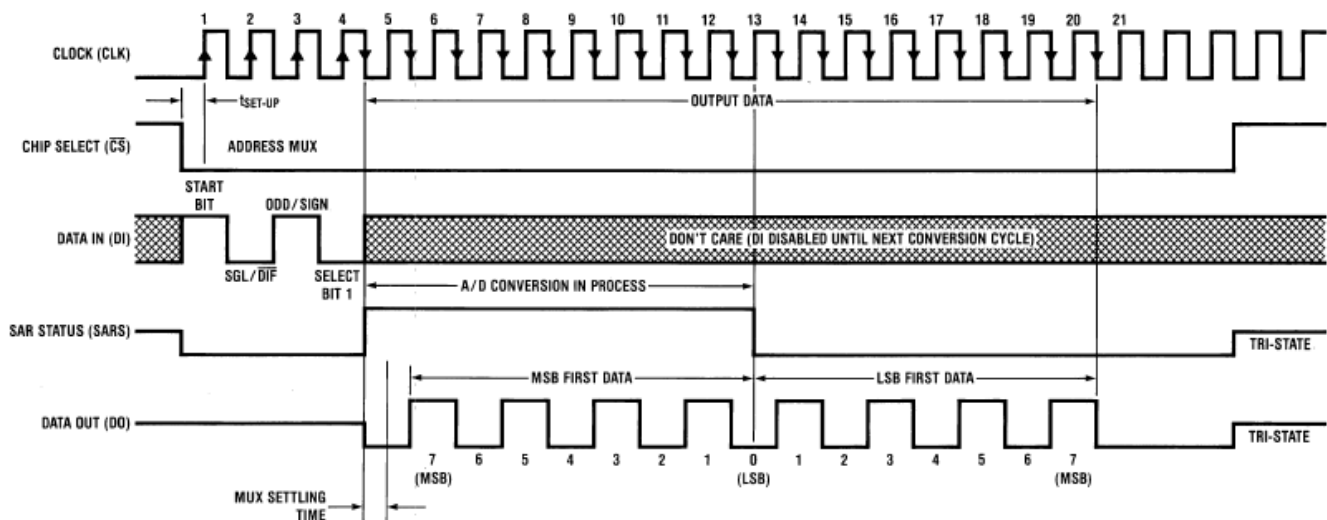


Figure 20. ADC0834-N Timing

رد الفعل اللي ال circuit بتعمله نتيجة القرايات كالاتي :

١. لما قراءة ال gas sensor تكون أكبر من 200ppm او درجة الحرارة اعلى من ٢٧ درجة سيلسي
 زيوس او درجة الرطوبة كانت اكبر من ٧٠ فان المروحة هتشتغل لمدة ١٢٠ ثانيه ويتم ارسال الرساله
 النصيه : the fan is turned on للرقم المخزن في الكود وترجع تتأكد مره أخرى من القرايات اذا كانت
 تعدلت ولا لا

٢. ثاني رد فعل يتعلق بال moist sensor عندما تكون قرايته اكبر من ١٧٠ فهذا يدل على وجود درجة
 قليله من الماء وبالتالي تعمل ال pump تلقائيا على اقصى جهد وهو ٥ فولت ويتم ارسال الرساله النصيه
 the pump is turned on : وتعمل check بعد ١٢٠ ثانيه



وكل هذه القيم نتيجة تجارب عمليه تم التأكد منها ان القيم مناسبه للنبات ولا يتم غمر الماكيت بالماء او تشغيل المروحة بلا داعي

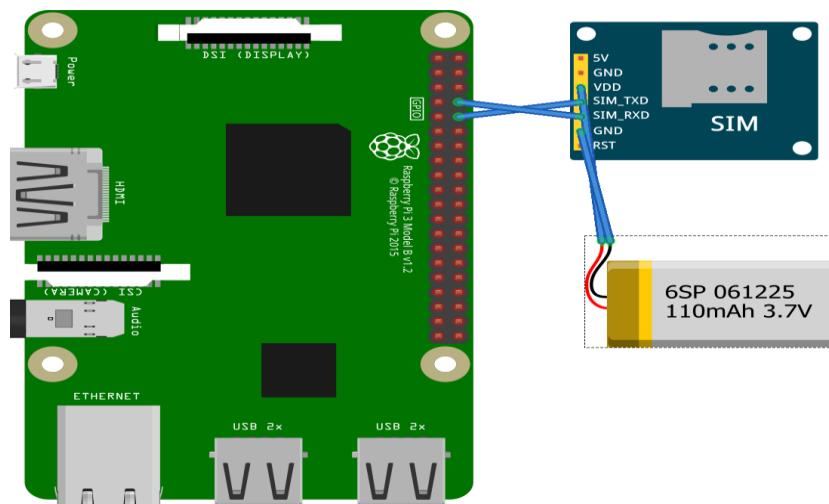
التحكم عن بعد باستخدام ال GSM 800I

ال GSM800I module بداخيه sim ذات رقم محدد وال sim تقوم بتخزين الرسائل النصيه في memory اقصى مدى لها من ٣٢ الى ١٢٨ كيلو بايتس وبداخل ال GSM مجموعه من ال registers اللتي تستقبل أوامر محدده يفهما ال GSM مثل قم باتصال او قم بتشغيل البيانات او قم بارسال رساله نصيه او ارسال أوامر محدده

تم استخدام

communication protocol: mini UART (Universal Assynchronous RX TX that uses start and end bits every time it RX and Tx)

حيث يستخدم في التواصل مع ال raspberry pi



fritzing

تم استخدام الدوال الاساسيه الاتيه وذلك للتواصل بين البوردتين :

1. `sim800l=SIM800L('/dev/ttyS0')` #Open port with baud rate(bits/sec)
2. `_sms=sim800l.read_sms(i)` #raspberrypi serial port is Rx info from GSM

اثناء مرحلة checking GSM memory

يتم اللجوء للمكتبه واستخدام المكتبه الاساسيه :

`command(self, cmdstr, lines=1, waitfor=500, msgtext=None):`

حيث ال argument الثاني يمثل عدد السطور التي ستخزن في ال pi والتي ستظهر على الشاشة في ال PI

١. حيث انه عندما يرغب ال pi في ارسال رساله لموبيل ذا رقم محدد نستخدم الامر :

`self.command('AT+CMGS="{0}"\n'.format(destno),99,5000,msgtext+'\x1A')`

٢. لما نحتاج نقرأ رساله واصله لل GSM فاحنا بنقرأ من ال memory الخاصه بال SIM فقط وتم وضع أقصى عدد من الرسائل في ال memory ٣ وبعد كده بتتشال

`result = self.command('AT+CMGR="{0}"\n'.format(id),99)`



```
ATE1
OK
AT+CMGF=1
OK
AT+CMGS="+370600000000"
> Sample text
>
+CMGS: 228
OK
```

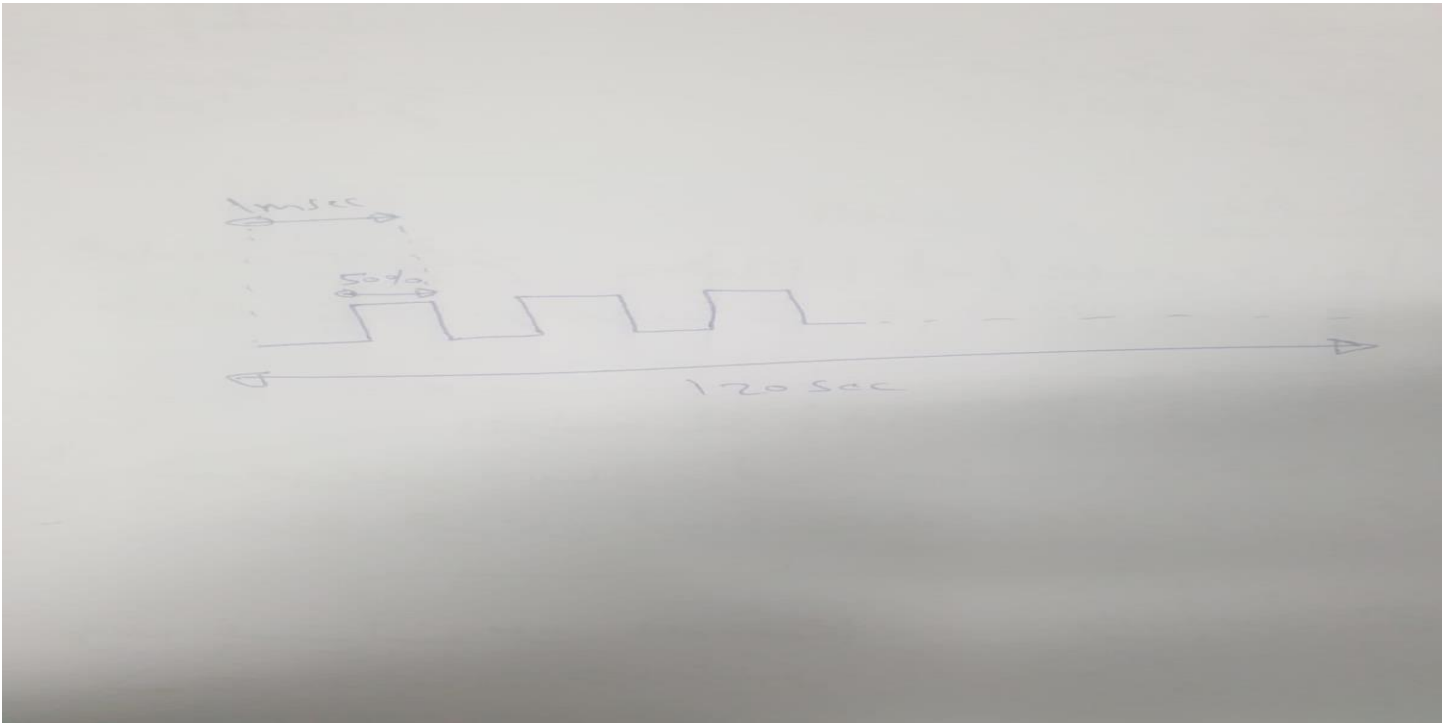
دي مجموعه من الأوامر اللي لما بكتبها على ال gsm بمن ال pi ال gsm بيرد علي OK

دي ليستة من ال commands اللي موجوده في ال data sheet الخاصه بال GSM800L:

Command	Description
AT+CMGD	DELETE SMS MESSAGE
AT+CMGF	SELECT SMS MESSAGE FORMAT
AT+CMGL	LIST SMS MESSAGES FROM PREFERRED STORE
AT+CMGR	READ SMS MESSAGE
AT+CMGS	SEND SMS MESSAGE
AT+CMGW	WRITE SMS MESSAGE TO MEMORY
AT+CMSS	SEND SMS MESSAGE FROM STORAGE
AT+CMGC	SEND SMS COMMAND
AT+CNMI	NEW SMS MESSAGE INDICATIONS
AT+CPMS	PREFERRED SMS MESSAGE STORAGE
AT+CRES	RESTORE SMS SETTINGS
AT+CSAS	SAVE SMS SETTINGS
AT+CSCA	SMS SERVICE CENTER ADDRESS
AT+CSCB	SELECT CELL BROADCAST SMS MESSAGES
AT+CSDH	SHOW SMS TEXT MODE PARAMETERS
AT+CSMP	SET SMS TEXT MODE PARAMETERS
AT+CSMS	SELECT MESSAGE SERVICE

ال gsm يستلم الأوامر الآتية في صورة رساله نصيه:

١. $p10 < p50$: حيث انه تم استخدام pulse width modulation للتحكم في خرج ال pump حسب الرسالة النصية حيث تكون ال duty cycle اكبر من 50% من ال time duration والذي يساوي 10^{-3} ثانيه والزمن الكلي هو ١٢٠ ثانيه



٢. $f <$ واللي يقوم بتشغيل المروحة لمدة ١٢٠ ثانيه

٣. $c <$ حيث يقوم هذا الامر بطلب من ال pi بالقيام بعمل recording لمدة ١٠ ثواني ومن ثم ارسال الملف لامييل معين

وتم استخدام مجموعه من ال objects الموجوده في ال raspberrypi وهي :

Message object: holds the body, recipient, transmitter and header

Attachment object: holds the path, the name and the extension of the picture or video

MIMEBase Object: encodes and encrypts the picture or the video

SMTP Server: sends the email to the recipient