

Rejoignez GitHub aujourd'hui

S'inscrire

Copier le chemin

 $\frac{1}{4}$

```
24 * connectez le capteur comme suit:
25 * Broche 2 du capteur de poussière PM1 -> Digital 3 (PMW)
26 * Broche 3 du capteur de poussière -> + 5V
27 * Broche 4 du capteur de poussière PM2.5 -> Digital 6 (PWM)
28 * Broche 5 du capteur de poussière -> Terre
29 * Fiche technique: http://www.samyongsnc.com/products/3-1%20Specification%20DSM501.pdf
30 * Contributeur: epierre
31 * * /
32
33 // Activer les impressions de débogage
34 # définir MY_DEBUG
35
36 // Activer et sélectionner le type de radio attaché
37 # define MY_RADIO_RF24
38 // #define MY_RADIO_NRF5_ESB
39 // #define MY_RADIO_RFM69
40 // #define MY_RADIO_RFM95
41
42 # include < MySensors.h >
43
44 # define CHILD_ID_DUST_PM10 0
45 # define CHILD_ID_DUST_PM25 1
46 # define DUST_SENSOR_DIGITAL_PIN_PM10 6
47 # define DUST_SENSOR_DIGITAL_PIN_PM25 3
48
49 uint32_t SLEEP_TIME = 30 * 1000 ; // Temps de sommeil entre les lectures (en millisecondes)
50 // VARIABLES
51 int val = 0 ; // variable pour stocker la valeur provenant du capteur
52 float valDUSTPM25 = 0,0 ;
53 float lastDUSTPM25 = 0.0 ;
54 float valDUSTPM10 = 0,0 ;
55 float lastDUSTPM10 = 0.0 ;
56 uint32_t duration;
57 uint32_t starttime;
58 uint32_t endtime;
59 uint32_t sampletime_ms = 30000 ;
60 uint32_t lowpulseoccupancy = 0 ;
61 rapport de flottement = 0 ;
62 longue concentrationPM25 = 0 ;
63 longue concentrationPM10 = 0 ;
64
65 MyMessage dustMsgPM10 (CHILD_ID_DUST_PM10, V_LEVEL);
66 MyMessage msgPM10 (CHILD_ID_DUST_PM10, V_UNIT_PREFIX);
67 MyMessage dustMsgPM25 (CHILD_ID_DUST_PM25, V_LEVEL);
68 MyMessage msgPM25 (CHILD_ID_DUST_PM25, V_UNIT_PREFIX);
69
70 void setup ()
71 {
72   pinMode (DUST_SENSOR_DIGITAL_PIN_PM10, INPUT);
73   pinMode (DUST_SENSOR_DIGITAL_PIN_PM25, INPUT);
74 }
75
```

```
76  présentation vide ()
77  {
78      // Envoyer les informations de version de l'esquisse à la passerelle et au contrôleur
79      sendSketchInfo ( " Capteur de poussière DSM501 " , " 1.4 " );
80
81      // Enregistre tous les capteurs sur la passerelle (ils seront créés en tant que périphériques)
82      present (CHILD_ID_DUST_PM10, S_DUST);
83      send (msgPM10. set ( " ppm " ));
84      present (CHILD_ID_DUST_PM25, S_DUST);
85      send (msgPM25. set ( " ppm " ));
86  }
87
88  boucle vide ()
89  {
90
91      // obtenir une densité de particules PM 2,5 supérieure à 2,5 µm.
92      concentrationPM25 = ( long ) getPM (DUST_SENSOR_DIGITAL_PIN_PM25);
93      En série. imprimer ( " PM25: " );
94      En série. impression (concentration PM25 );
95      En série. print ( " \ n " );
96
97      if ((concentrationPM25 != lastDUSTPM25) && (concentrationPM25 > 0 )) {
98          send (dustMsgPM25. set (( int32_t ) ceil (concentrationPM25)));
99          lastDUSTPM25 = ceil (concentrationPM25);
100     }
101
102     // PM 1.0 - densité de particules supérieure à 1 µm.
103     concentrationPM10 = getPM (DUST_SENSOR_DIGITAL_PIN_PM10);
104     En série. imprimer ( " PM10: " );
105     En série. impression (concentrationPM10);
106     En série. print ( " \ n " );
107     // ppmv = mg / m3 * (0.08205 * Tmp) / Molecular_mass
108     // 0,08205 = Constante universelle des gaz en atm · m3 / (kmol · K)
109     int temp = 20 ; // température externe, si vous pouvez remplacer ceci par un DHT11 ou un DS18B20
110     long ppmv = (concentrationPM10 * 0,0283168 / 100 / 1000 ) * ( 0,08205 temp *) / 0,02446;
111
112     if (( ceil (concentrationPM10) != lastDUSTPM10) && (( long ) concentrationPM10 > 0 )) {
113         send (dustMsgPM10. set (( int32_t ) ppmv));
114         lastDUSTPM10 = ceil (concentrationPM10);
115     }
116
117     // dormir pour économiser à la radio
118     dormir (SLEEP_TIME);
119
120 }
121
122
123 getPM long ( int DUST_SENSOR_DIGITAL_PIN)
124 {
125
126     heure de début = millis ();
127
```

```
128   tandis que ( 1 ) {
129
130       duration = pulseIn (DUST_SENSOR_DIGITAL_PIN, LOW);
131       faible occupation + = durée;
132       heure de fin = millis ();
133
134       if ((endtime-starttime)> samptime_ms) {
135           ratio = (lowpulseoccupancy-endtime + starttime) / (samptime_ms * 10.0 ); // Po
136           concentration longue = 1,1 * pow (ratio, 3 ) - 3,8 * pow (ratio, 2 ) + 520 * ratio;
137           // Serial.print ("lowpulseoccupancy:");
138           // Serial.print (lowpulseoccupancy);
139           // Serial.print ("\ n");
140           // Serial.print ("ratio:");
141           // Serial.print (ratio);
142           // Serial.print ("\ n");
143           // Serial.print ("DSM501A:");
144           // Serial.println (concentration);
145           // Serial.print ("\ n");
146
147           faible occupation du vide = 0 ;
148           retour (concentration);
149       }
150   }
151 }
```