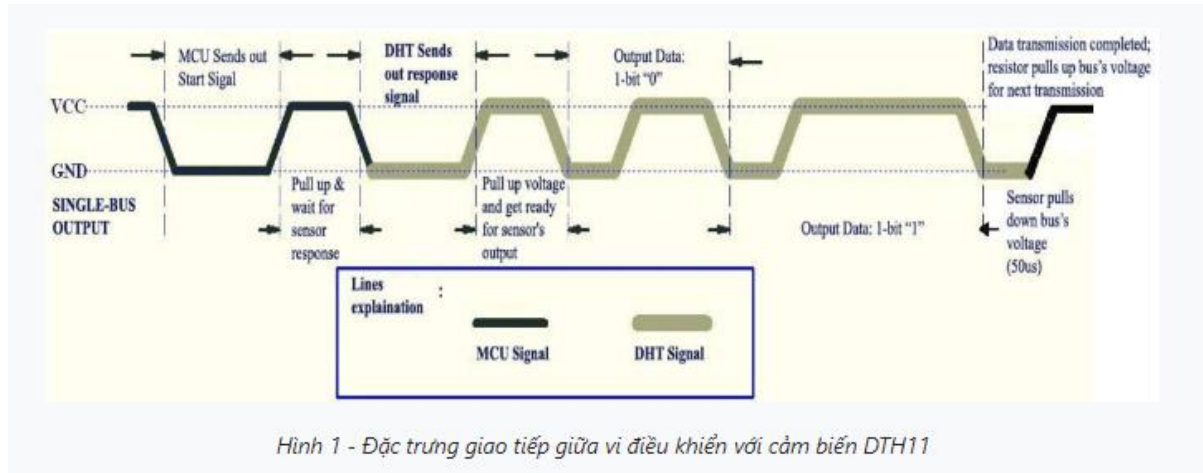


3. Nội dung:

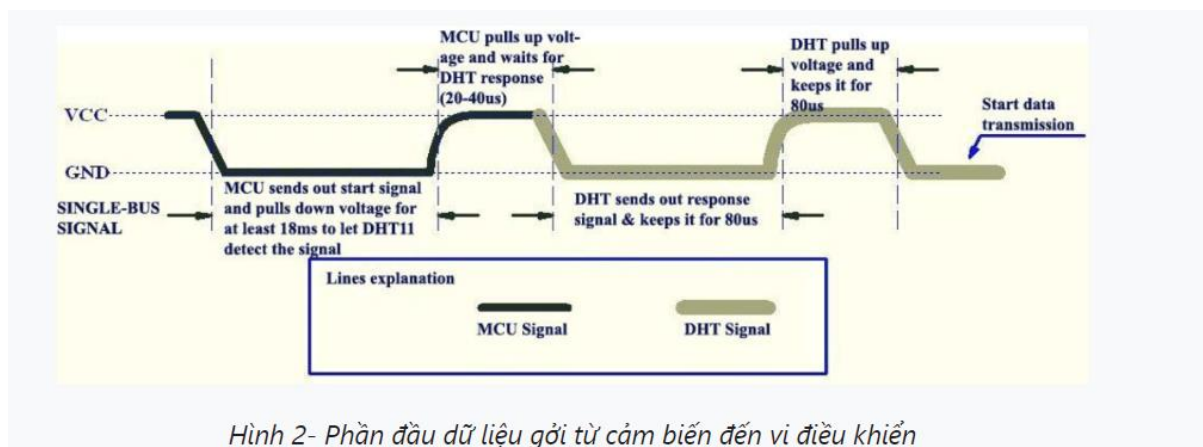
3.1. Phân tích dự án:

- Cần làm theo datasheet của nhà sản xuất để đọc được giá trị cảm biến



- Đầu tiên Vi điều khiển gửi 1 xung (đường màu đen) lên chân Data tới cảm biến. Sau đó chờ nhận dữ liệu nối tiếp theo quy định sau (protocol). Cảm biến gửi lại dữ liệu theo thứ tự : Phần bắt đầu -> Phần dữ liệu.

- Phần bắt đầu gồm: 1 xung mức thấp 80us rồi đến mức cao 80us, theo sơ đồ sau:



- Phần dữ liệu theo thứ tự sau:

- Data consists of decimal and integral parts. A complete data transmission is 40bit, and the sensor sends higher data bit first.
- Data format: <8bit integral RH data> + <8bit decimal RH data> + <8bit integral T data> + <8bit decimal T data> + <8bit check sum>.
- If the data transmission is right, the check-sum should be the last 8bit of "8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data".

- Với các mức logic 0 và logic 1 được nhận diện như sau: (logic 0 có chiều dài mức cao từ 26-28us; logic 1 có chiều dài mức cao từ 70us)

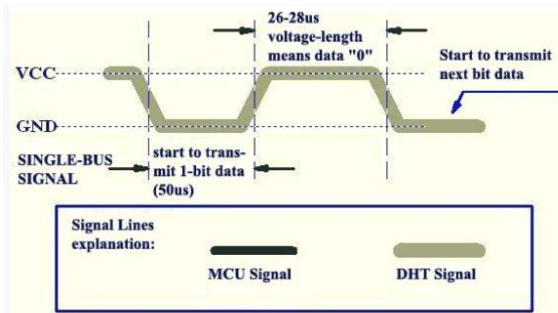


Figure 4 Data "0" Indication

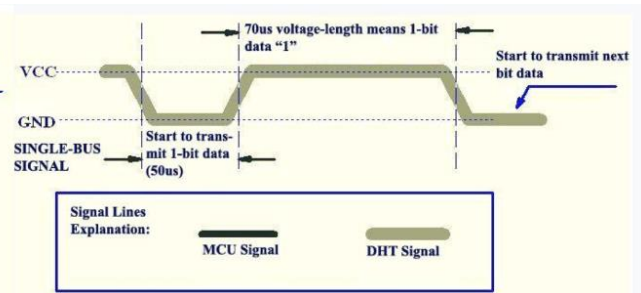
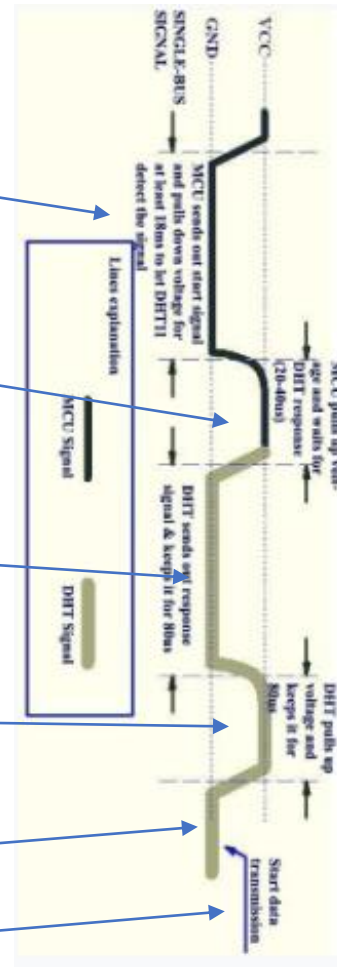
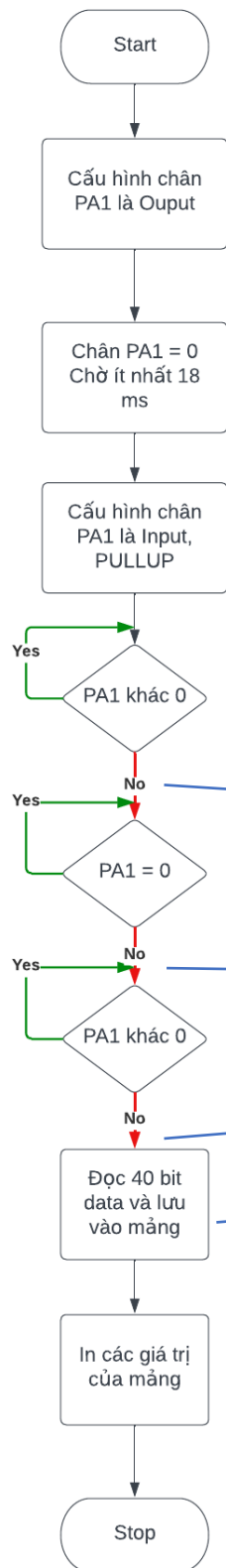
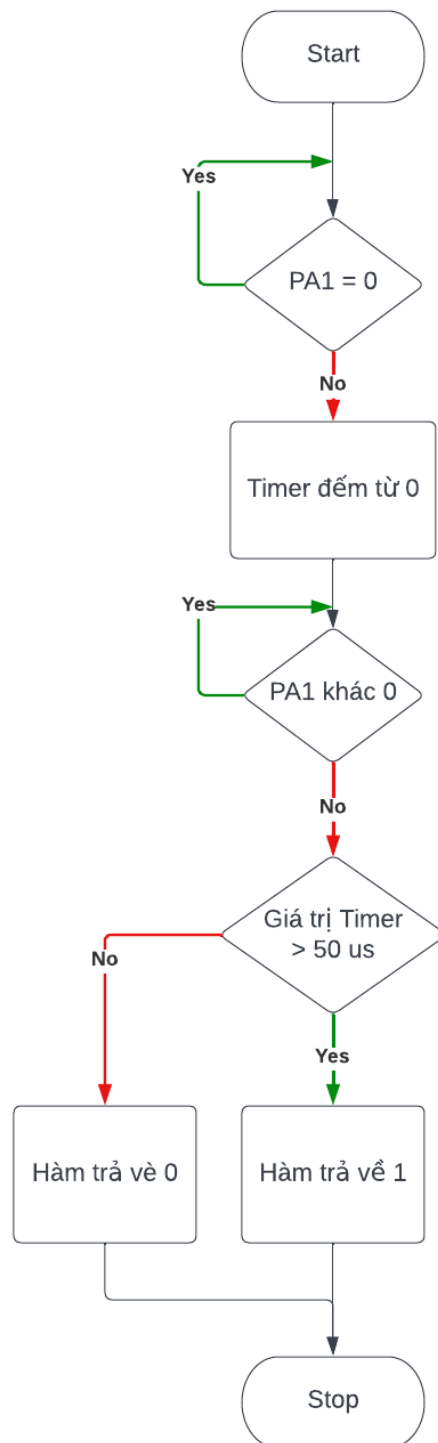


Figure 5 Data "1" Indication

Hình 3 - Đặc trưng logic 0 và logic phát ra từ cảm biến

3.2. Lưu đồ lập trình:





Hàm nhận dạng mức logic

3.3. Mã nguồn chương trình:

- Tạo hàm printf để in kết quả qua UART

```
51 /* USER CODE BEGIN EV */
52 FILE __stdout;
53 int fputc(int ch, FILE *f)
54 {
55     HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, HAL_MAX_DELAY);
56     return ch;
57 }
58 /* USER CODE END EV */
```

- Tạo hàm nhận diện mức logic bằng cách so sánh thời gian Timer đo được

```
70 /* USER CODE BEGIN 0 */
71 uint8_t OneWire_ReadBit(void) {
72     #define DELAY50US 200
73     while(HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_1)==0);
74     __HAL_TIM_SET_COUNTER(&htim1,0);
75     while(HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_1)!=0);
76     if (__HAL_TIM_GET_COUNTER(&htim1) > DELAY50US)
77         return 1;
78     else
79         return 0;
80 }
81
```

- Trong vòng lặp chính

- Cấu hình chân PA1 là output. Cho chân PA1 = 0 và chờ 19ms

```
121 while (1)
122 {
123     /* USER CODE END WHILE */
124     /*Configure GPIO pin : PA1 */ // pull_up
125     GPIO_InitStruct.Pin = GPIO_PIN_1;
126     GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
127     GPIO_InitStruct.Pull = GPIO_PULLUP;
128     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
129     HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
130     HAL_GPIO_WritePin(GPIOA,GPIO_PIN_1,GPIO_PIN_RESET);
131     HAL_Delay(19);
132 }
```

- Cấu hình chân PA1 là Input (PULL_UP). Đợi chân PA1 tạo 1 cạnh xuống, rồi tạo 1 cạnh lên, rồi tạo 1 cạnh xuống (--_--_)

```
132
133     GPIO_InitStruct.Pin = GPIO_PIN_1;
134     GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
135     GPIO_InitStruct.Pull = GPIO_PULLUP;
136     HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
137     while (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1) != 0);
138     while (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1) == 0);
139     while (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1) != 0);
140
```

- Gán dữ liệu 40 bit data đọc được từ cảm biến (đọc bằng hàm nhận diện mức logic) vào mảng

```
140
141     uint8_t DHT_Data[] = {0,0,0,0,0};
142     for (uint8_t i=0; i<5; i++) {
143         for (uint8_t j=0; j<8; j++) {
144             DHT_Data[i] = DHT_Data[i] | (OneWire_ReadBit() << (7-j));
145         }
146     }

```

- In các phần tử của mảng tương ứng với giá trị độ ẩm và nhiệt độ. Mỗi lần đo delay 500ms

```
148     printf("B0=%u;", DHT_Data[0]);
149     printf("B1=%u;", DHT_Data[1]);
150     printf("B2=%u;", DHT_Data[2]);
151     printf("B3=%u", DHT_Data[3]);
152     HAL_Delay(500);

```

4. Ghi chú khác (nếu có)