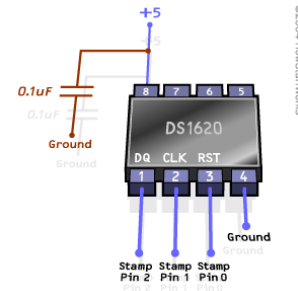


TP 3 - Mesure de la température ambiante



Objectifs :

- Développer un programme destiné à mesurer la température ambiante à travers une carte usb I/O.
- Utiliser des classes personnalisées de la carte Velleman P8055.
- Analyser la notice technique constructeur afin de concevoir une nouvelle classe objet DS1620_P8055 dérivée de la classe P8055.

But :

- Réaliser l’affichage d’un thermomètre digital sur PC grâce au circuit DS1620 interconnecté au module K8055 de Velleman lui-même connecté sur port USB du PC.

Critères d'évaluation:

Vous serez évalués sur les critères suivants:

- * Votre autonomie à résoudre les différents problèmes énoncés.
- * Votre capacité à mettre en oeuvre le travail étudié.
- * Votre capacité à utiliser le matériel mis à votre disposition.
- * La rédaction de l'algorithme et du programme de mise en oeuvre.

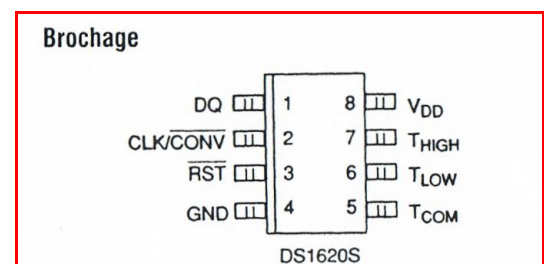
Travail demandé

Développer une nouvelle classe DS1620_P8055, dérivée de la classe P8055 pour permettre l’élaboration d’un programme en C++ obtenant et affichant, en temps réel, la température en degré Celsius de votre environnement.

1) Etude de la notice constructeur:

notice constructeur: DS1620.pdf

D0	entrée/sortie de l'interface trifilaire
Clk/Conv\	entrée de l'horloge trifilaire et entrée de conversion autonome
RST	entrée de RAZ trifilaire
GND	masse
Thigh	déclencheur de température haute
Tlow	déclencheur de température basse
Tcom	déclencheur de combinaison high/low
VDD	tension d'alimentation positive (+5 V)



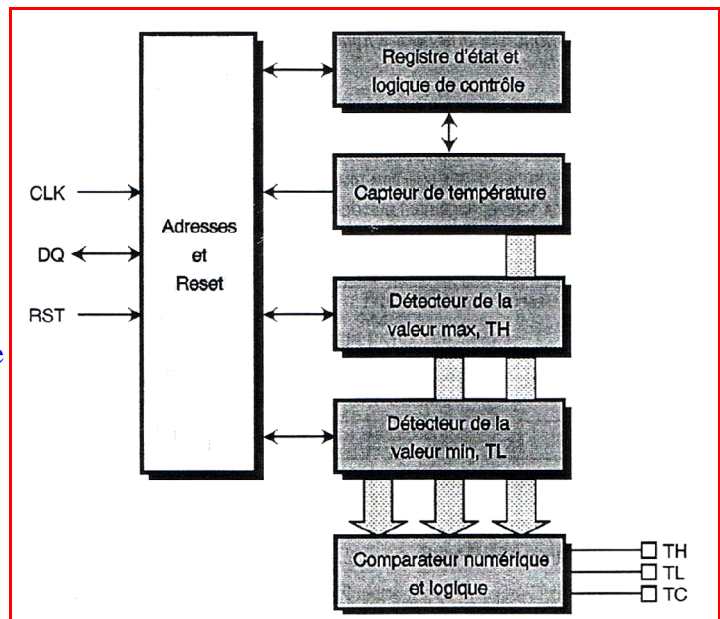
Description

Le thermomètre numérique et thermostat DS1620 fournit une indication de température sur 9 bits qui rend compte de la température du dispositif lui-même. Grâce à trois sorties de commande, le DS1620 peut aussi servir de thermostat. Thigh sera positionné à l'état haut si la température du DS1620 est supérieure ou égale à la température Th définie par l'utilisateur. Tlow sera positionné à l'état haut si la température du DS1620 est inférieure ou égale à la température TL définie par l'utilisateur. Tcom sera positionné à l'état haut quand la température dépassera TH et restera à l'état haut jusqu'à ce que la température descende en dessous de TL.

Les valeurs de température définies par l'utilisateur sont stockées dans une mémoire non volatile, de sorte que les composants peuvent être programmés avant leur mise en place dans un système, de même qu'ils peuvent être utilisés dans des applications autonomes sans processeur. Les paramètres de température et les valeurs de température relevées sont tous échangés avec le DS1620 via une simple interface à trois fils.

Caractéristiques :

- Ne nécessite aucun composant externe
- Capable de mesurer des températures de -55°C à $+125^{\circ}\text{C}$ par pas de $0,5^{\circ}\text{C}$
- Résolution en température de 9 bits
- Convertit la température en numérique en 1 seconde.
- Paramètres du thermostat définissables par l'utilisateur et non volatiles
- Les données sont lues/écrites par interface série à trois fils (CLK, Da, ST)
- Le champ d'application couvre le contrôle thermostatique, les systèmes industriels, les produits grand public, les thermomètres ou tous systèmes sensibles à la chaleur.
- Boîtiers DIP à 8 broches ou SOIC



Le dialogue entre le DS 1620 et le monde extérieur pour programmer la température de consigne ainsi que pour lire la température ambiante s'effectue par trois broches DQ, CLK-/CONV\ . La température est fournie sur 9bits en complément à deux.

Le tableau suivant indique la relation entre les données en sortie et la température mesurée en degré Celsius; pour connaître la température en degré Fahrenheit, une table de conversion doit être utilisée. Les données sont transmises en sortie sous forme série par la broche DQ, le LSB en premier (bit de poids le plus faible, c'est à dire d0).

Température en sortie	Donnée Binaire	Donnée Hexa
+125°C	0 1111 1010	00FA
+25°C	0 0011 0010	0032
+0,5°C	0 0000 0001	0001
0°C	0 0000 0000	0000
-0,5°C	1 1111 1111	01FF
-25°C	1 1100 1110	01CE
-55°C	1 1001 0010	0192

Tout transfert de données est initialisé en imposant un niveau haut sur la broche RST; la mise à zéro de cette dernière termine la communication et la sortie DQ est alors en haute impédance.

Un cycle d'horloge est composé par un front descendant suivi par un front montant sur la broche CLK.

Les données sont disponibles en sortie sur la broche DQ au front descendant de l'horloge, et restent valides jusqu'au front montant suivant. Puis, pendant que l'horloge est au niveau haut, la sortie DQ est en haute impédance.

Pour écrire dans le circuit, les données doivent être valides durant le front montant d'un cycle d'horloge.

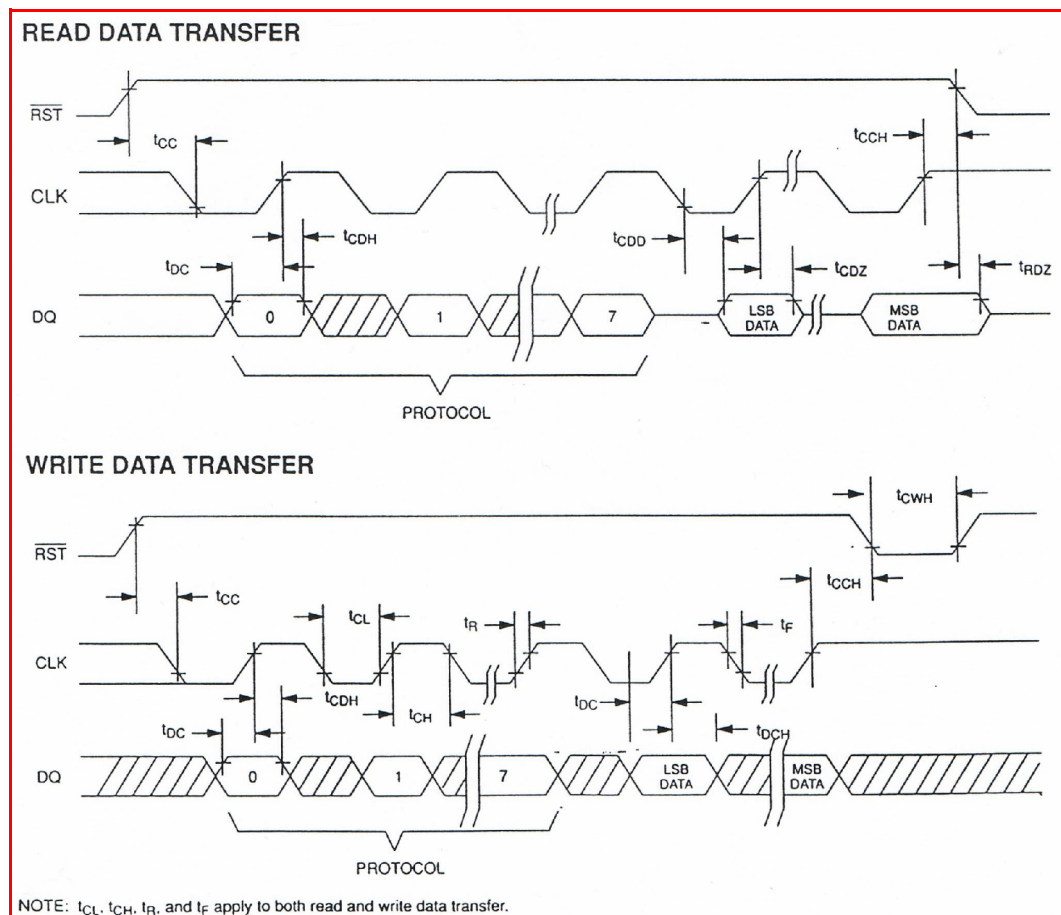
Pour transmettre des commandes au DS 1620, on écrit des données en série sur la broche DQ, le LSB étant envoyé en premier.

Pour le mode thermomètre, seules trois commandes nous intéressent:

Read Temperature [AAH] : Cette commande lit le contenu du registre qui contient le résultat de la dernière mesure de température. Les neuf cycles d'horloge suivants permettront de lire le contenu de ce registre.

Start Convert [EEH] : Cette commande démarre une conversion en 1 seconde de la température. Aucun autre envoi n'est nécessaire.

Stop Conversion [22H] : Cette commande stoppe la conversion de la température, et le DS 1620 reste en attente d'une nouvelle commande de début de conversion.

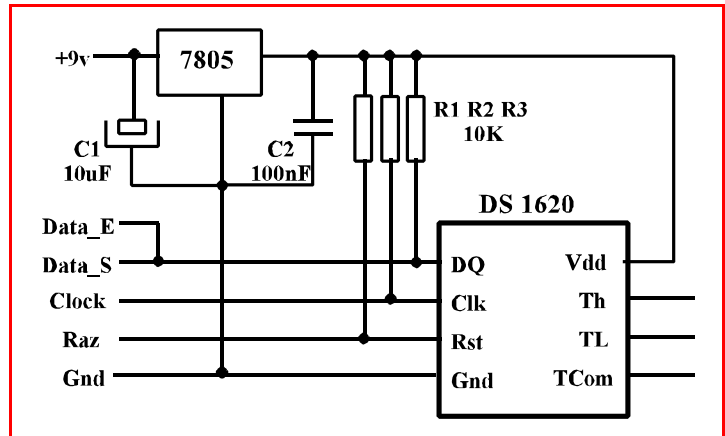


- Donner les principales caractéristiques du circuit (alimentation, type de liaison, etc...) .
- Donner et analyser les différentes broches de commande.
- Relever les temps caractéristiques minimales pour la transmission de données.
- Déterminer clairement le protocole pour obtenir une lecture de température.

2) Montage:

Le circuit est câblé sur une carte d'essai, il est alimenté par un régulateur 7805.

L'ensemble est relié à la carte Velleman P8055 tel que :



Carte d'essai	Gnd	Data_S	Data_E	Clock	Raz
Velleman K8055	Gnd	Q1	I1	Q2	Q3

- Relier le montage à la carte Velleman K8055 et alimenter l'ensemble à une alimentation stabilisée de 9v.

3) Programme:

- Créer une nouvelle classe "**DS1620_P8055**" qui dérive de la classe "**P8055**".

Pour cela , créer, sous Qt Creator un nouveau projet "Application graphique Qt, ajouter les fichier .h et .cpp de la classe P8055 puis créer la nouvelle classe **DS1620_P8055** contenant les méthodes suivantes:

- en mode private :

void clock () ; // fait passer l'entrée Clock à 1 puis le remet à 0 pour une validation de la donnée du circuit DS1620.

void protocol (char val) ; // génère les signaux clock et data du DS1620 pour envoyer une instruction 8 bits protocole.

- en mode public:

DS1620_P8055 (int addcarte) ; // constructeur qui passe les paramètres au constructeur de la classe mère pour initialiser la carte K8055 et qui met à '1' l'entrée clock et à '0' l'entrée RAZ.

~DS1620_P8055 () ; // destructeur

void lance_conv () ; // génère les signaux à l'aide des méthodes décrites précédemment pour lancer une conversion de température du DS1620.

void stop_conv () ; // génère les signaux à l'aide des méthodes décrites précédemment pour stopper la conversion de température du DS1620.

short Lect_Temp () ; // génère les signaux de commande pour lire le registre contenant le dernier résultat issu de la conversion de température du DS1620.

- Modifier votre application Qt permettant de tester toutes les méthodes de la classe "**DS1620_P8055**" et qui affiche la température, en temps réel, chaque seconde.

