Technologieschema:

Beschreibung: Für eine Wetterstation sollen Temperaturmesswerte aufgezeichnet und ausgewertet werden.

– Messintervall: 10s – bis 20 Messwerte, weiter Messwerte überschreiben den Speicher wieder von Anfang an. – Aktueller Messwert in Zeile 1 anzeigen – Messwertauswahl für die Anzeige mit PA6 und PA10 – Messwerte liegen digitalisiert an Ain0 vor. – Anzeige von Minimum und Maximum in der 2. Zeile – Anzeige des mit den Tasten *vorheriger* und *nächster* ausgewählten Temperatur in der 1. Zeile des

Displays – *vorheriger* scrollt im Messwertspeicher pro Tastendruck um einen Messwert abwärts bis zum

Messwert Nr 0 und bringt diesen zur Anzeige – *nächster* scrollt im Messwertspeicher pro Tastendruck um einen Messwert aufwärts bis zum letzten

aufgezeichneten Messwert und bringt diesen zur Anzeige

Aufgabe:

– Portzuordnungsliste – Was bedeutet A/D? In welchem Wertebereich liegen die Temperaturen

In welchem Speicher wird aufgezeichnet? Welche Instruktionen sind erforderlich?

– Programm in ASM

PA6

PA10

D

A

Ain0

nächster

Messwert anzeigen vorheriger

μController

Temperatur

LCD-Anzeige

Lösungshinweise: RAM verwenden: Das RAM nimmt den Adressbereich von 0x20000000 bis 0x20013FFF ein. Zugriffsmöglichkeiten: ldr R1,=Messwerte //gewünschte RAM-Adresse laden ldrb R2,[R1,0] //ein Byte aus dem RAM lesen Adresse 0x20000000 und in R2 eintragen strb R3,[R1,1] //ein Byte aus R3 im RAM an der Adresse 0x20000001 abspeichern

mov R4,#5 strb R3,[R1,R4] //speichert ein Byte aus R3 in der Adresse R1+R4 also 0x20000005 add R4,#1 ldrb R5,[R1,R4] //lädt ein Byte aus R1+R4 also 0x20000006 in R5

Am Programmende: (vor end) ... **.data** //Datenbereich im RAM Messwerte: .space 20 //reserviert 20 Bytes entspricht: char Messwerte[20];

.end

oder: ... **.data** //Datenbereich im RAM Messwerte: .byte 20,25,35,17 //reserviert 4 Bytes mit den vorbelegten Werten 25,25,35,17

//entspricht: char Messwerte[4]={20,25,35,17}; .end

AD-Wandler: call Ain0 //überträgt den Einstellwert des Potis nach R0

Zeitschleife: mov R0,#1000 //1000ms = 1s bl wait\_ms