



## Einsendeaufgabe Typ B

## Grundlagen der Informatik mit Labor

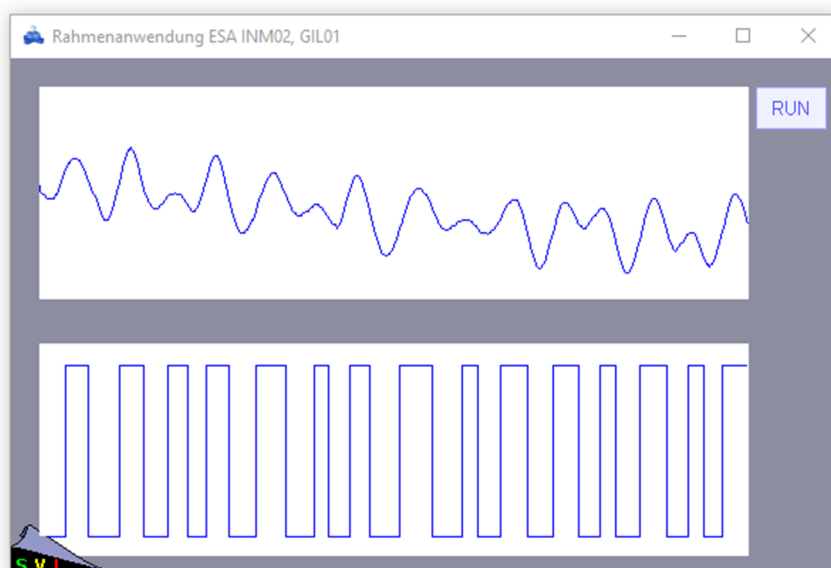
Name:		Vorname:	Einsendeaufgabencode: <b>B-GIL01-XX7-K02</b>
Straße:		PLZ, Ort:	Korrektor:
Matrikelnummer:		Studiengangsnummer:	Datum:
Name der B-Aufgabe: <b>B-G I L 0 1 X X</b>	Variante: <b>7</b>	Auflage: <b>1222K02</b>	Note:
Bezogene Studienhefte: <b>SEI11, INM01/INM02 oder INM11/INM12, INM-L</b>			Unterschrift:

Bitte reichen Sie zu Ihren Lösungen zwingend auch die Aufgabenstellung und den Einsendeaufgabencode mit ein – alles zusammen über den Online-Campus.

Senden Sie die Lösung für Programmieraufgaben in einer Form ein, die es dem Gutachter ermöglicht diese mit einem Compiler zu übersetzen, also als maschinenlesbaren Text (ANSI-Text-Format) oder als SiSy-Projekt-Archiv (Menüpunkt: Projekt/Archiv/Archiv anlegen).

Alle nicht maschinenlesbaren Formen können leider nicht bewertet werden.

1. Die Erfassung von Messwerten ist oft mit Störsignalen, Oberwellen oder Grundschrwingungen beaufschlagt. Eine wichtige Teilaufgabe besteht darin, aus einem solchen Signal ein ursprüngliches Impulssignal zu rekonstruieren. Gehen Sie davon aus, dass Sie Teil eines größeren Entwicklungsteams sind und hier eine bestimmte Teilaufgabe zu lösen haben. Die Signalerfassung und weitere Verarbeitung wird durch andere Teammitglieder realisiert.



**Abb.01:** Lösung der Teilaufgabe in einer Test-Rahmenanwendung

Ihre Teilaufgabe wird durch die Festlegung einer verbindlichen Schnittstelle (Funktionssignatur) gekapselt. Die von Ihnen zu entwickelnde Funktion wird als digitaler Filter in einem großen und unter Umständen sehr schnellen kontinuierlichen Datenstrom bei jedem einzelnen gemessenen Wert aufgerufen. Sie haben keinen Zugriff auf den gesamten Datenstrom, sondern immer nur auf den letzten gemessenen Wert. Für die Realisierung solcher Teilaufgaben werden oft Rahmenanwendungen mit Testklassen zur Verfügung gestellt. Die Abbildung zeigt das Ergebnis der Lösung.

Entwickeln Sie eine Funktion für die Erkennung von Impulsflanken (steigende/fallende Flanken) mit folgender Signatur:

```
int edgeDetect(double value)
{
    int edge = -1; // Flankenerkennung ist ausgeschaltet
    // edge = 0 ... fallende Flanke erkannt
    // edge = 1 ... steigende Flanke erkannt
    // eine Flanke gilt als erkannt bei mindestens 5 bis 10
    // aufeinander folgenden fallenden oder steigenden Werten
    return edge;
}
```

Sie finden ein Musterprojekt in *INM02* bzw. *GIL01* für die Entwicklungsumgebung im **SiSy-LibStore** beim Anlegen eines SiSy-Projekts. Dieses Projekt enthält eine Rahmenanwendung mit Testsignalgenerator und Testdatenvisualisierung.

**40 Pkt.**

2. Gegeben ist folgender Textausschnitt des Lastenhefts:

*Die Steuerung soll den Füllstand im Mischbehälter der Anlage regeln. Die Steuerung kann vom Leitstand ein neues Mischverhältnis empfangen. Wenn ein neues Mischverhältnis vorliegt, muss die Steuerung die Pumpenlaufzeiten neu berechnen. Der Anlagenzustand wird nur auf Anforderung des Leitstands gesendet. Der Servicetechniker kann bei Bedarf die Software der Steuerung über eine USB-Schnittstelle aktualisieren.*

Dokumentieren Sie für das Pflichtenheft die Top-Level-Anforderungen an die Anlage mit den Sprachmitteln des UML-Anwendungsfalldiagramms (use case diagram).

**20 Pkt.**

3. Gegeben ist folgender Textausschnitt des Lastenhefts:

*Nach dem Start der Steuerung geht diese auf „Warten“. Während des Wartens wird fortlaufend der Füllstand des Mischbehälters gemessen. Wenn der Füllstand das Minimum erreicht hat, geht die Steuerung auf „Leer“. In diesem Moment werden die Pumpen eingeschaltet, dann wird fortlaufend der Füllstand überwacht. Wenn der Füllstand das Maximum erreicht hat, geht die Steuerung auf „Voll“. Dabei müssen die Pumpen ausgeschaltet werden. Nach einer Wartezeit von einer Sekunde (Time Out) geht die Steuerung wieder auf „Warten“.*

Dokumentieren Sie die funktionalen Anforderungen an die Anlage mit den Sprachmitteln des UML-Zustandsdiagramms (state machine).

**20 Pkt.**

4. Erstellen Sie ein UML-Sequenzdiagramm für den folgenden C++-Code:

```
void MainWnd::onBtnSend()
{
    if (steuerung.isConnected())
    {
        dataFrame = mixControl.getData();
        steuerung.write(pilotBytes);
        steuerung.write(dataFrame);
    }
    else
    {
        this->messageBox("keine Verbindung");
    }
}
```

**20 Pkt.**

**Gesamt: 100 Pkt.**