جامعة الموصل / كلية الهندسة / قسم هندسة الحاسوب

مختبرات المستوى الرابع

وصف مختبر الزمن الحقيقي

1. المعلومات العامة:	
اسم المختبر و رقم المختبر:	مختبر الزمن الحقيقي (مختبر 211)
اسم المقرر المرتبط:	أنظمة الزمن الحقيقي
القسم:	هندسة الحاسوب
عدد الساعات الأسبوعية للمختبر:	3 ساعات
عدد أسابيع الفصل الدراسي:	15 أسبوعًا
المستوى الدراسي:	المستوى الرابع
المشرف على المختبر:	د. بسمان محمود الحافظ

2. وصف عام للمختبر:

يمختبر الزمن الحقيقي هو أحد مختبرات قسم هندسة الحاسوب التابع لكلية الهندسة في جامعة الموصل. يهتم بدراسة تجارب الزمن الحقيقي وتصميم و تحليل دوائرها. يخدم المختبر طلبة الدراسات الأولية والدراسات العليا كما يوفر الأجهزة اللازمة لإجراء البحوث العملية لتدريسيي القسم المهتمين بهذا المجال.

3. أهداف المختبر:

- حل المشاكل والتحليل: تطبيق مبادئ الهندسة والعلوم والرياضيات لتحديد وتحليل وحل المشاكل المعقدة.
- التعلم المستمر: اكتساب المعرفة الجديدة باستخدام استراتيجيات مناسبة وإدراك الحاجة للتطوير المهني المستمر.
- التعاون المهني: العمل بأخلاقية وفعالية في بيئات الفرق متعددة التخصصات.
- أساسيات أنظمة الزمن حقيقي: فهم المكونات الصلبة والبرمجية، تصنيف الأنظمة، أجهزة الاستشعار، تكييف الإشارات، ناقلات البيانات (GPIB, RS232)، وأجهزة التخزين.
- تطوير البرمجيات لانظمة الزمن حقيقي: تصميم وتنفيذ برمجيات التحكم التي تلبي متطلبات التوقيت ومواجهة تحديات أنظمة الزمن الحقيقي.

4. مخرجات التعلم:

سيتمكن الطلاب من:

1. **تصميم أنظمة الزمن الحقيقي**: تطوير التطبيقات باستخدام منصات Arduino و Raspberry Pi مع تكامل المستشعر.
2. **تحليل خصائص المستشعر**: تقييم وتنفيذ أجهزة استشعار مختلفة لتطبيقات العالم الحقيقي.

3. **تطبيق برمجة Python**: استخدم Python لتطوير الأنظمة المضمنة ومعالجة البيانات.
4. **تصميم مشاريع كاملة:** تخطيط وتنفيذ وتقديم حلول الأنظمة المضمنة من المفهوم إلى التنفيذ.
5. **العمل بشكل تعاوني**: العمل بفعالية في بيئات الفريق وتوصيل الحلول التقنية

5. الجدول الأسبوعي للتجارب:

الأسبوع	عنوان التجربة	الأدوات / البرامج المستخدمة	الهدف الرئيسي
1	Introduction	Datashow	التعرف على مفاهيم عامة عن أنظمة الزمن حقيقي
2	Sensor Characteristics part 1	Sensor kit &kl31001 board	لدراسة خصائص وعمل متحسسات متنوعة
3	Sensor Characteristics part 2	Sensor kit &kl31001 board	لدراسة خصائص وعمل متحسسات متنوعة
4	Arduino sensors – Module 1, Module 2, Module 3	Arduino board and sensor	التعرف على الاردوينو و متحسساته
5	Example of real-world applications	Arduino board & esp32 and sensor	التعرف على كيفية تصميم أنظمة زمن حقيقي
6	Review and quiz 1	-----	مراجعة عامة وتقييم
7	Introduction to Python Language	Datashow	التعرف على اساسيات لغة python
8	Intro. to Raspberry Pi H.W. Computer	Raspberry Pi	التعرف على raspberry pi وكيفية برمجته
9	Raspberry Pi - Sense HAT Part 1	Raspberry Pi & Sense HAT	كيفية استخدام raspberry pi مع المتحسسات
10	Raspberry Pi - Sense HAT Part 2	Raspberry Pi & Sense HAT	كيفية استخدام raspberry pi مع المتحسسات
11	Review and quiz 2	All components needed	مراجعة عامة وتقييم
12	Project Implementation1	All components needed	تقييم عملي لتنفيذ مشاريع المجموعة
13	Project Implementation2	All components needed	تقييم عملي لتنفيذ مشاريع المجموعة
14	Projects Discussions – Part 1	Datashow	مناقشة المشاريع العملية وتقييمها

15	Projects Discussions – Part 2	Datashow	مناقشة المشاريع العملية وتقييمها

6. الأدوات والمعدات المستخدمة:

- Raspberry pi 4
- K&H Sensors kit
- KL-31001
- Raspberry pi 3
- Sense HAT
- Arduino mega
- Avometer
- Oscilloscope
- Desktop Computer

7. دليل السلامة:

- التأكد من فصل التيار الكهربائي عند توصيل الأجهزة.
- عدم لمس المنافذ الكهربائية أو مكونات الشبكة دون إذن المشرف.
- الالتزام بالهدوء وتنظيم الكابلات لتجنب الحوادث.
- استخدام البرنامج المحاكي لتجارب التوجيه قبل التجريب على الأجهزة الحقيقية.

8. آلية التقييم:

عنصر التقييم	النسبة
الحضور والمشاركة	12%
الاختبارات القصيرة	20%
الامتحان العملي النهائي	40%
المشروع العملي	28%

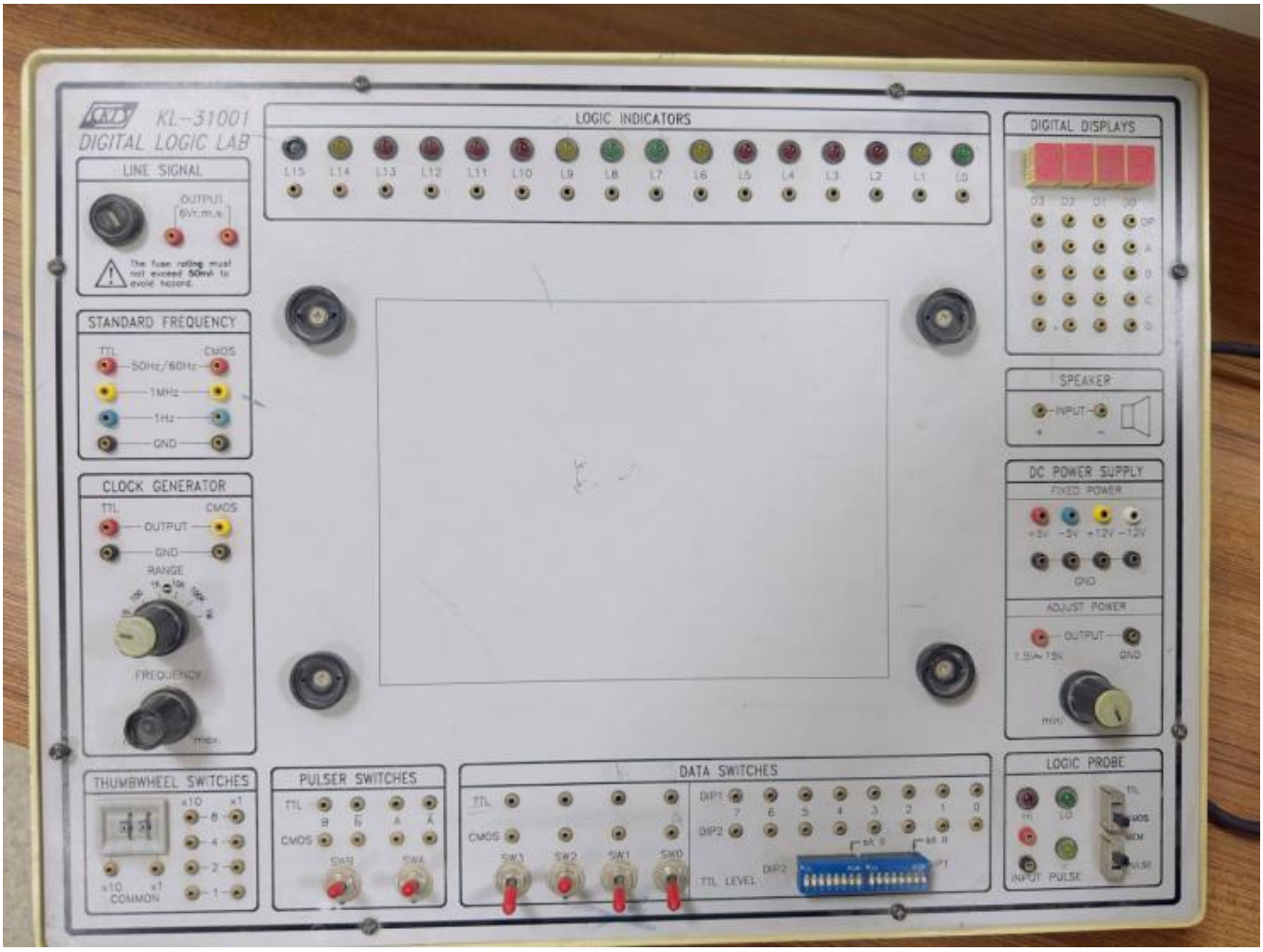
9. المراجع والمصادر:

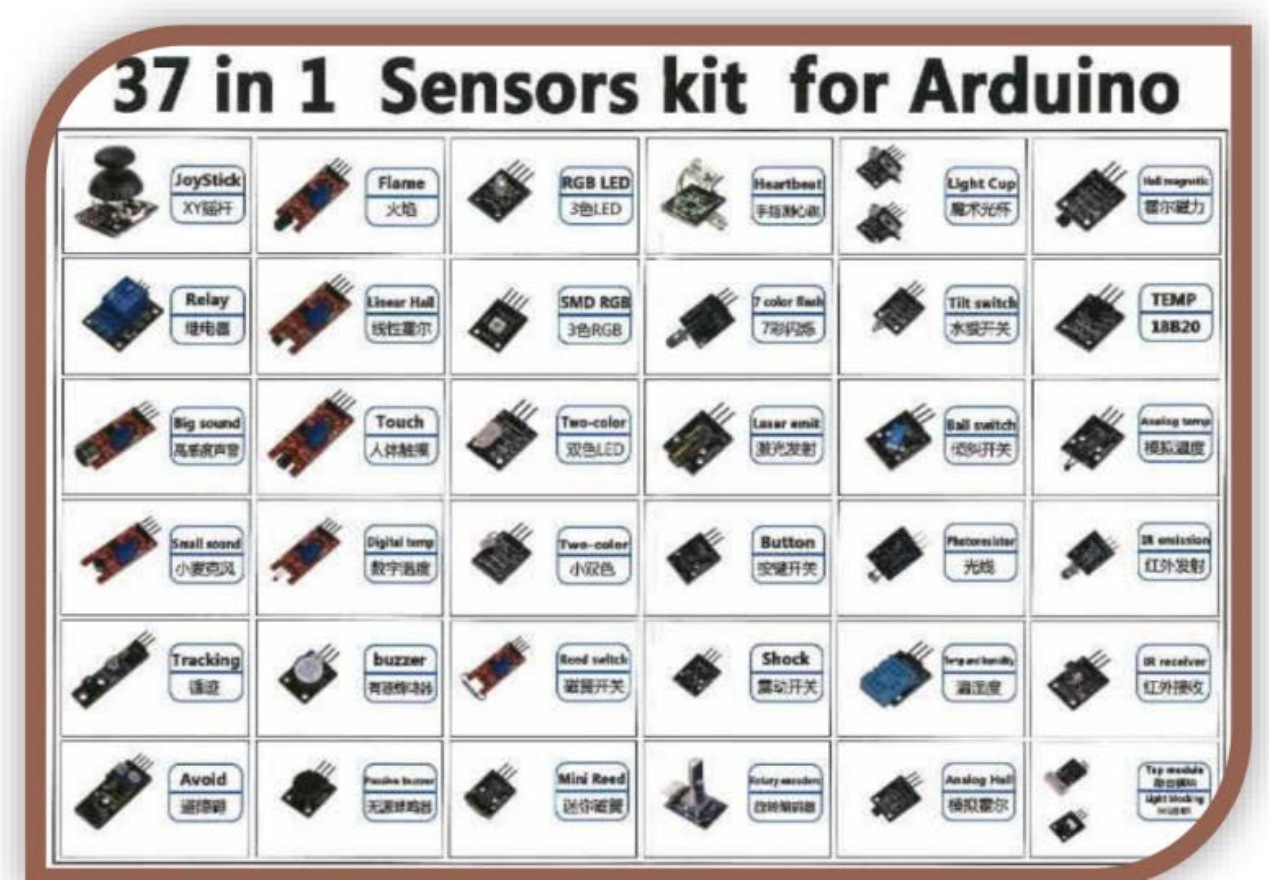
- K&H Documents

- استخدام برنامج Arduino IDEو Trinker
- electronicshub.org-getting started with esp32 introduction to esp32
- مواقع تعليمية مثل https://www.raspberrypi.com, Arduino.com, anaconda.com/

10.المرفقات:

- خطة العمل لتجارب الفصل وطريقة التقييم
- مستوى_الخطورة_لمختبر الزمن الحقيقي _قسم_هندسة_الحاسوب

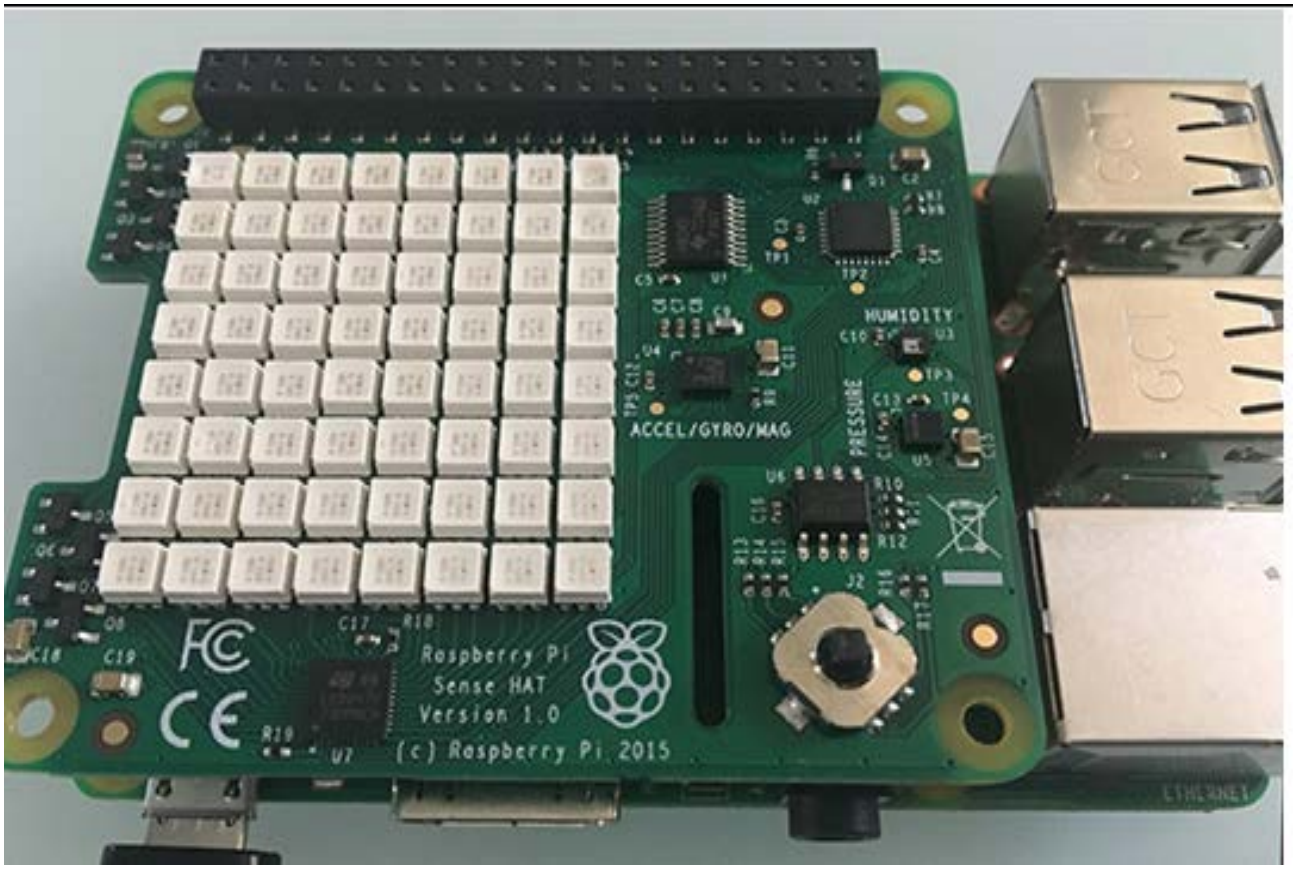

KL-31001
DIGITAL LOGIC LAB
LOGIC INDICATORS
DIGITAL DISPLAYS
LINE SIGNAL
STANDARD FREQUENCY
CLOCK GENERATOR
SPEAKER
DC POWER SUPPLY
ADJUST POWER
THUMBWHEEL SWITCHES
PULSER SWITCHES
DATA SWITCHES
LOGIC PROBE

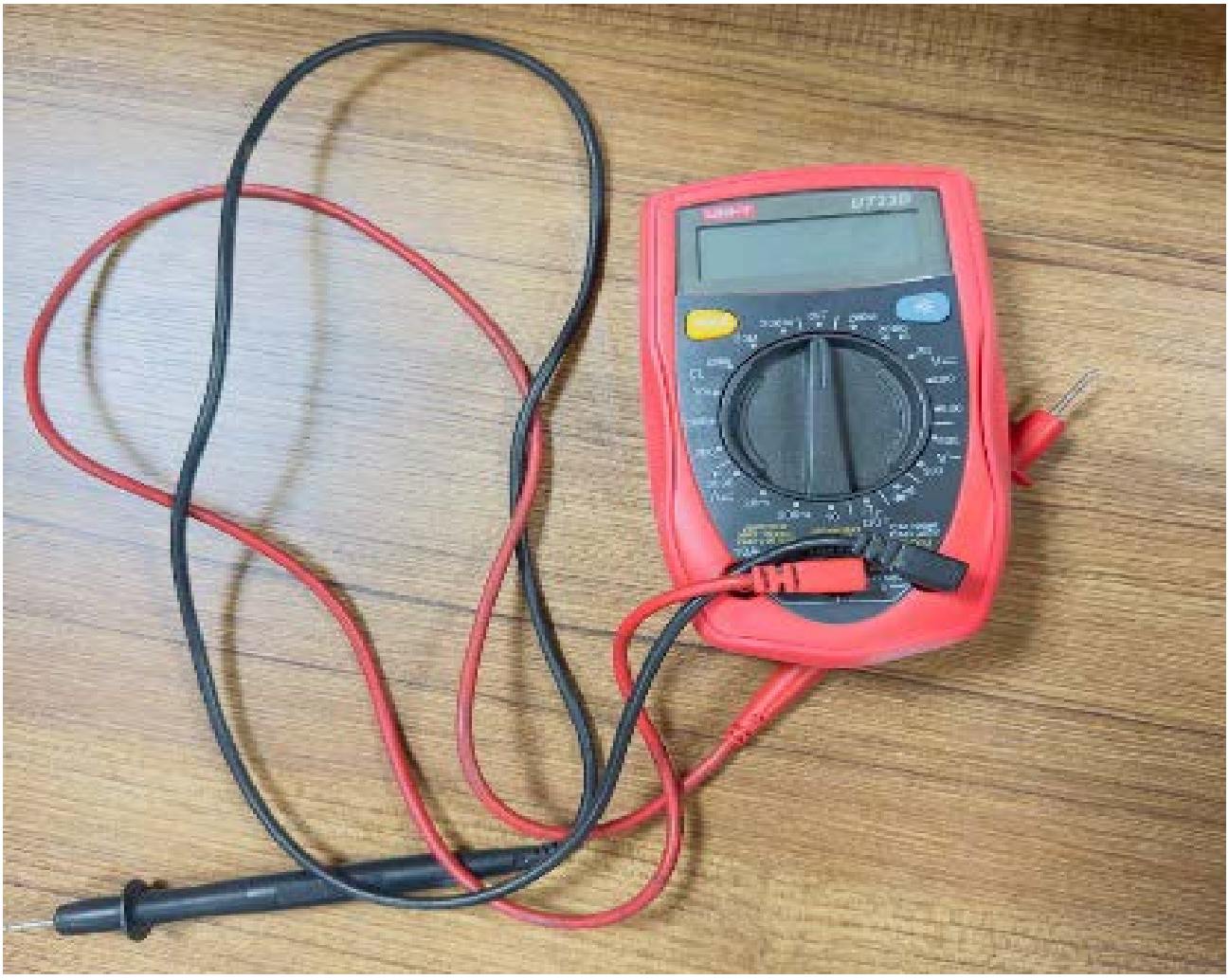

37 in 1 Sensors kit for Arduino
JoyStick
Flame
RGB LED
Heartbeat
Light Cup
Relay
Linear Hall
SMD RGB
Tilt switch
TEMP 18B20
Big sound
Touch
Two-color
Laser emit
Ball switch
Small sound
Digital temp
Button
Photoresister
Tracking
buzzer
Reed switch
Shock
Avoid
Mini Reed
Analog Hall

KL-64003
SENSOR MODULE 3

KL-64002
SENSOR MODULE 2

Raspberry Pi 3


HUMIDITY
PRESSURE
ACCEL/GYRO/MAG
Raspberry Pi
Sense HAT
Version 1.0
(c) Raspberry Pi 2015



وصف مختبر أنظمة السيطرة (مختبر 210)

1. المعلومات العامة:	
اسم المختبر و رقم المختبر:	مختبر أنظمة السيطرة
اسم المقرر المرتبط:	أنظمة السيطرة
القسم:	هندسة الحاسوب
عدد الساعات الأسبوعية للمختبر:	4 ساعات
عدد أسابيع الفصل الدراسي:	15 أسبوعًا
المستوى الدراسي:	المستوى الرابع
المشرف على المختبر:	د. سرى نوفل

2. وصف عام للمختبر:

يُوفر المختبر للطلاب خبرة عملية في تحليل وتصميم وتنفيذ أنظمة التحكم. ويغطي مجموعة واسعة من المواضيع، بدءًا من برمجة MATLAB الأساسية وصولًا إلى تقنيات التحكم المتقدمة، بما في ذلك أنظمة التحكم التناظرية والرقمية. يدمج المختبر المعرفة النظرية مع التطبيقات العملية، باستخدام أدوات مثل MATLAB وSimulink وأجهزة التحكم المنطقية القابلة للبرمجة (PLCs) والحواسيب التناظرية.

3. أهداف المختبر:

يهدف هذا المختبر إلى سد الفجوة بين المعرفة النظرية والتطبيقات العملية من خلال سلسلة من التجارب العملية. وتتمثل الأهداف الرئيسية في:

- تعريف الطلاب بالأدوات الأساسية مثل MATLAB و Simulinkلتحليل أنظمة التحكم.
- توفير خبرة عملية في تصميم وتنفيذ أنظمة التحكم باستخدام الطرق التناظرية والرقمية.
- تعريف الطلاب بأنظمة التحكم في الوقت الفعلي ووحدات التحكم المنطقية القابلة للبرمجة.(PLCs)
- تطوير مهارات تحليل استجابات النظام، وضبط وحدات التحكم، والتعامل مع تحديات التحكم.
- تشجيع العمل الجماعي وحل المشكلات من خلال مجموعة تصميم أنظمة التحكم.

4. مخرجات التعلم:

بنهاية الدورة، سيُظهر الطلاب قدرتهم على تصميم وتنفيذ أنظمة التحكم، بما في ذلك وحدات تحكم PID وبرمجة PLC. سيُطبقون المفاهيم النظرية لضبط استجابات النظام، وتحسين الأداء، واستكشاف تحديات التحكم الواقعية. سيُطور الطلاب مهارات دمج الأجهزة والبرمجيات لبرمجة أنظمة التحكم اللحظية باستخدام أجهزة استشعار ذكية. تُؤهلهم هذه الكفاءات للمهام الهندسية المتقدمة وتطبيقات الأتمتة الصناعية (مخرج 2).

5. الجدول الأسبوعي للتجارب:			
الأسبوع	**عنوان التجربة**	**الأدوات / البرامج المستخدمة**	**الهدف الرئيسي**
1	أساسيات ماتلاب لأنظمة السيطرة	MATLAB	التعرف على MATLAB، وهي أداة أساسية لتحليل وتصميم أنظمة السيطرة.
2	واجهة المستخدم الرسومية (GUI) باستخدام MATLAB	MATLAB	بناءً على أساسيات MATLAB، يتعلم الطلاب كيفية إنشاء واجهات سهلة الاستخدام لتطبيقات التحكم.
3	أساسيات التحكم وتخفيض الكتلة باستخدام MATLAB.	MATLAB	فهم مفاهيم التحكم الأساسية وتطبيق MATLAB لتقنيات تقليل الكتلة.
4	دالة النقل والتحويل بين تمثيلات النظام	MATLAB	استكشاف وظائف النقل والتحويل بين تمثيلات النظام المختلفة.
5	استجابة الخطوة وخطأ الحالة المستقرة.	MATLAB	تطبيق تحويلات لابلاس في MATLAB لتحليل استجابة الخطوة ودراسة خطأ الحالة المستقرة.
6	أجهزة الكمبيوتر التناظرية	MULTISIM+ MATLAB	استكشاف عملي لأجهزة الكمبيوتر التناظرية وأهميتها في محاكاة ديناميكيات نظام التحكم.
7	وحدة تحكم PID باستخدام أجهزة الكمبيوتر التناظرية	MULTISIM+ MATLAB LOGO Siemens Simulator	تصميم مكونات PID باستخدام RLC ومكبرات التشغيل
8	برمجة PLC	LOGO Siemens Simulator	مقدمة إلى وحدات التحكم المنطقية القابلة للبرمجة والخبرة العملية في البرمجة.
9	برمجة PLC (مخططات Ladder خاصة)	LOGO Siemens Simulator	مقدمة لبرمجة وتوصيل مخططات السلم الخاصة
10	تطبيقات PLC	LOGO Siemens Simulator	التطبيقات الواقعية لأجهزة التحكم المنطقية القابلة للبرمجة في سيناريوهات التحكم، مع التركيز على حالات الاستخدام الصناعية.

11	طريقة التصميم باستخدام Bode Plot	MATLAB	فهم استجابة التردد من خلال تحليل مخطط Bode.
12	اتصال LabVIEW LINX (Raspberry PI)	LabVIEW LINX+ **Raspberry PI**	ربط LabVIEW بالأجهزة الخارجية Raspberry Pi باستخدام LabVIEW LINX.
13	تصميم وضع الأعمدة Pole Placement	MATLAB, LabVIEW	تطبيق تقنية وضع الأقطاب لتحقيق الأداء المطلوب للنظام.
14	مراجعة شاملة	MATLAB, LabVIEW, all Hardware Interface	القيام بمراجعة جميع التجارب وتقييم الفهم العملي والأداء العام.
15	الامتحان الفصلي	الامتحان الفصلي	الامتحان الفصلي

6. الأدوات والمعدات المستخدمة:

البرامج:

- MATLAB (مجموعة أدوات نظام السيطرة، Simulink)
- LabVIEW (وحدة تصميم التحكم والمحاكاة، وحدة الزمن الحقيقي)
- Multisim (أو محاكي الدائرة التناظرية المكافئ)
- محاكي LOGO Siemens (برنامج PLC)

الأجهزة:

- راسبيري باي
- أجهزة كمبيوتر تناظرية / مجموعات تدريب
- أجهزة تحكم منطقية قابلة للبرمجة (PLC) (إن وجدت) - أجهزة تحكم منطقية قابلة للبرمجة من سيمنز
- مصادر طاقة
- أجهزة قياس (منظار ذبذبات ومقاييس متعددة رقمية)
- معدات مختبر الإلكترونيات العامة: ألواح التجارب، والأسلاك، والموصلات، والوصلات

معدات المختبر العامة:

- ألواح التجارب ولوحات النماذج الأولية
- الكابلات والموصلات
- متحسسات القياس

7. دليل السلامة:

اتبع دائمًا تعليمات المدرب قبل تشغيل أي جهاز أو معدة.

- تأكد من التوصيل الصحيح لجميع واجهات الأجهزة قبل التشغيل.
- تأكد من صحة تكوين أجهزة LabVIEW Real-Time وDAQ لتجنب تلف النظام.
- افصل مصادر الطاقة قبل تعديل أي دائرة.
- تعامل مع جميع أجهزة الكمبيوتر وأجهزة جمع البيانات ووحدات التحكم بحرص لتجنب أي تلف مادي أو كهربائي.
- أبلغ مسؤل المختبر فورًا عن أي سلوك غير طبيعي للنظام (إخراج غير متوقع، ارتفاع درجة الحرارة، رسائل خطأ).
- حافظ على نظافة وترتيب مساحة العمل لمنع انقطاع الكابلات العرضي أو سقوط المعدات.
- اتبع جميع بروتوكولات السلامة العامة في المختبر.

8. آلية التقييم:

عنصر التقييم	النسبة
الحضور والمشاركة	13%
اختبار عملي	13%
أداء المختبر	27%
تقارير الزيارات العلمية	13%
الاختبار العملي الفصلي	34%

9. المراجع والمصادر:

- نايز، ن. س. هندسة أنظمة التحكم، الطبعة السابعة، وايلي.
- أوغاتا، ك. هندسة التحكم الحديثة، الطبعة الخامسة، برنتيس هول.
- مستندات /MATLAB: https://www.mathworks.com/help/matlab
- مستندات مجموعة أدوات تصميم التحكم LabVIEW: https://www.ni.com
- دليل مستخدم LabVIEW، ناشيونال إنسترومنتس.
- Nise, N. S. *Control Systems Engineering*, 7th Edition, Wiley.
- Ogata, K. *Modern Control Engineering*, 5th Edition, Prentice Hall.
- MATLAB Documentation: https://www.mathworks.com/help/matlab/
- LabVIEW Control Design Toolkit Documentation: https://www.ni.com
- LabVIEW User Manual, National Instruments.

10.المرفقات:

- جدول المختبر:
- شرح عن إجراءات التجربة(sheet)
- نماذج ملفات LabVIEW VI
- نماذج نصوص MATLAB:

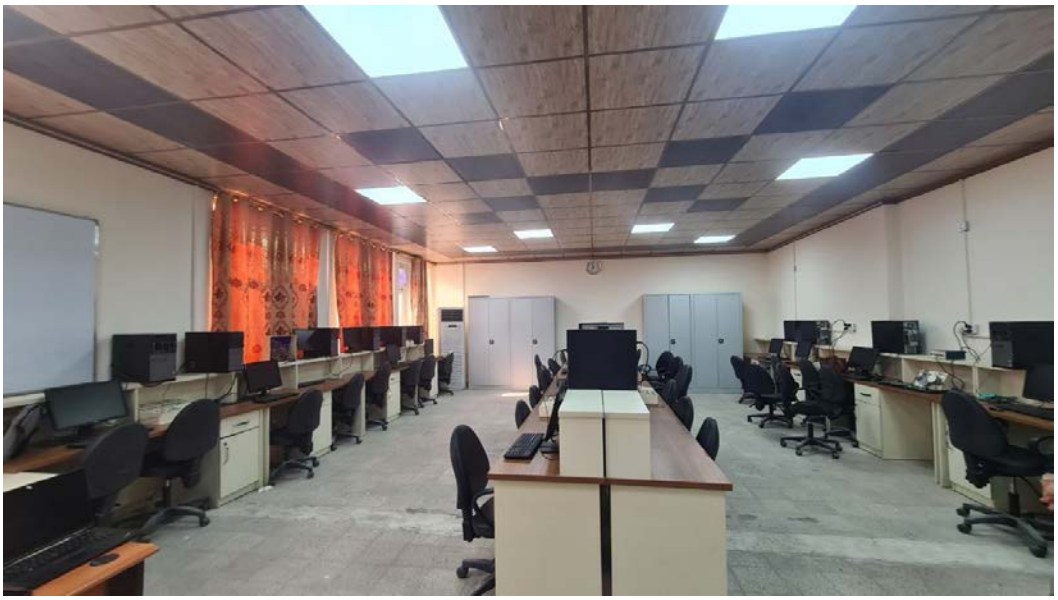
Laboratory tools instruments

Analogue Servo unit 110-33 , Mechanical unit control and instrumentation 100-33.

Data Acquisition Cards.

Raspberry Pi 4 and 3 versions. Arduino and sensors set.

	LOGO 12\24 REC PLC.



وصف مختبر السيطرة الرقمية (مختبر 210)

1. المعلومات العامة:	
اسم المختبر و رقم المختبر:	مختبر سيطرة رقمية
اسم المقرر المرتبط:	سيطرة رقمية
القسم:	هندسة الحاسوب
عدد الساعات الأسبوعية للمختبر:	4 ساعات
عدد أسابيع الفصل الدراسي:	15 أسبوعًا
المستوى الدراسي:	المستوى الرابع
المشرف على المختبر:	د. سرى نوفل

2. وصف عام للمختبر:

يوفر هذا المختبر خبرة عملية في تحليل وتصميم ومحاكاة وتنفيذ أنظمة التحكم. تدمج التجارب أدوات برمجية مثل MATLAB وLabVIEW، وتغطي مفاهيم السيطرة الأساسية والمتقدمة، بما في ذلك السيطرة الكلاسيكية، ونمذجة فضاء الحالة(state space) ، والسيطرة الرقمية، والتنفيذ في الزمن الحقيقي. يهدف المختبر إلى ربط المعرفة النظرية بالتطبيقات العملية باستخدام المحاكاة والواجهات الرسومية.

3. أهداف المختبر:

يهدف هذا المختبر إلى سد الفجوة بين المعرفة النظرية والتطبيقات العملية من خلال سلسلة من التجارب العملية. وتتمثل الأهداف الرئيسية في:

- توفير خبرة عملية في نمذجة ومحاكاة وتصميم أنظمة السيطرة الرقمية.
- تعريف الطلاب بالأدوات القياسية في هذا المجال: MATLAB وLabVIEW.
- تطوير مهاراتهم في طرائق السيطرة التقليدية (Nyquist ,Bode, Root locus).
- تقديم مواضيع متقدمة مثل فضاء الحالة state space ، والمترقبات observers، والسيطرة الرقمي.
- تنفيذ أنظمة سيطرة في الزمن الحقيقي باستخدام واجهات الأجهزة ووحدات LabVIEW Real-Time.

4. مخرجات التعلم:

حل مسائل أنظمة التحكم الرقمي باستخدام تحويل Z، ورسم مخطط محاكاة لأنظمة التحكم الرقمي،ومن ثم تصميم وتنفيذ وحدات سيطرة رقمية متنوعة باستخدام MATLAB للتحكم في محركات التيار المستمر والمحركات الخطوية steper motor. دمج وبرمجة أنظمة السيطرة بالزمن الحقيقي باستخدام أجهزة استشعار ذكية (مخرج 2).

5. الجدول الأسبوعي للتجارب:			
الأسبوع	عنوان التجربة	الأدوات / البرامج المستخدمة	الهدف الرئيسي
1	تحويل لابلاس باستخدام MATLAB	MATLAB	تحويل لابلاس باستخدام MATLAB
2	اساسيات الـLABVIEW	LabVIEW	تعريف ببيئة LabVIEW ووظائفها الأساسية.
3	نموذج تصميم عناصر السيطرة فيLABVIEW	LabVIEW Control Design Toolkit	تطبيق مبادئ تصميم عناصر التحكم باستخدام LabVIEW مع المحاكاة والنمذجة.
4	تصميم الـ root locus	Matlab	فهم وتطبيق طريقة root locus لتصميم أنظمة السيطرة.
5	معوضات root locus	Matlab/SISO tool	تصميم المعوضات باستخدام root locus لتحقيق أداء التحكم المطلوب.
6	تحليل استجابة الزمن باستخدام LabVIEW	LabVIEW	تحليل وتصور خصائص استجابة الزمن باستخدام LabVIEW.
7	تصميم وحدات السيطرة PID	LabVIEW, MATLAB	تصميم وحدات تحكم تناسبية تكاملية مشتقة (PID) باستخدام طريقة ضبط زيجلر-نيكولز.
8	التصميم باستخدام بيئة LabVIEW	LabVIEW	تطبيقات LabVIEW المتقدمة في تصميم أنظمة السيطرة.
9	ربط LabVIEW LINX	LabVIEW LINX+Arduino	ربط LabVIEW مع الأجهزة الخارجية مثل Arduino و Raspberry Pi باستخدام LabVIEW LINX
10	التحكم في المحركات باستخدام Arduino وLabVIEW	LabVIEW LINX+Arduino	خبرة عملية في التحكم بالمحركات باستخدام Arduino وLabVIEW
11	تصميم السيطرة الرقمية	MATLAB	مقدمة في أنظمة التحكم الرقمية ومبادئ التصميم.
12	وحدة السيطرة على محرك السائر باستخدام LabVIEW	LabVIEW+DAQ+stepper	ستخدام LabVIEW لاكتساب البيانات (DAQ) مع محركات السائر.

	عبر DAQ		
13	التحكم في محركات التيار المستمر باستخدام وحدة تحكم P	LabVIEW+DAQ+servo motor	تصميم وحدة تحكم تناسبية.
14	التحكم في محركات التيار المستمر باستخدام وحدة تحكم PID	LabVIEW+DAQ+servo motor	تنفيذ التحكم PID لأنظمة محركات التيار المستمر.
15	امتحان الفصل الدراسي العملي	امتحان الفصل الدراسي العملي	امتحان الفصل الدراسي العملي

6. الأدوات والمعدات المستخدمة:

البرامج:

- MATLAB (مجموعة أدوات نظام السيطرة، Simulink)
- LabVIEW (وحدة تصميم التحكم والمحاكاة، وحدة الزمن الحقيقي)
- برامج مسوق LabVIEW DAQmx
- مستكشف القياس والأتمتة من National Instruments (NI MAX)

الأجهزة:

- أجهزة اكتساب البيانات من National Instruments (NI USB-6009)
- وحدات مصادر الطاقة
- نماذج محركات التيار المستمر وآليات التشغيل
- أجهزة الاستشعار (أجهزة التشفير، مقاييس سرعة الدوران، مقاييس الجهد، ...)
- مولدات الإشارة
- راسمة الموجات

معدات المختبر العامة:

- ألواح التجارب ولوحات النماذج الأولية
- الكابلات والموصلات
- متحسسات القياس

7. دليل السلامة:

اتبع دائمًا تعليمات المدرب قبل تشغيل أي جهاز أو معدة.

• تأكد من التوصيل الصحيح لجميع واجهات الأجهزة قبل التشغيل.

• تأكد من صحة تكوين أجهزة LabVIEW Real-Time وDAQ لتجنب تلف النظام.

• افصل مصادر الطاقة قبل تعديل أي دائرة.

• تعامل مع جميع أجهزة الكمبيوتر وأجهزة جمع البيانات ووحدات التحكم بحرص لتجنب أي تلف مادي أو كهربائي.

• أبلغ مسؤل المختبر فورًا عن أي سلوك غير طبيعي للنظام (إخراج غير متوقع، ارتفاع درجة الحرارة، رسائل خطأ).

• حافظ على نظافة وترتيب مساحة العمل لمنع انقطاع الكابلات العرضي أو سقوط المعدات.

• اتبع جميع بروتوكولات السلامة العامة في المختبر.

8. آلية التقييم:

عنصر التقييم	النسبة
الحضور والمشاركة	13%
اختبار عملي	13%
أداء المختبر	27%
تقارير الزيارات العلمية	13%
الاختبار العملي الفصلي	34%

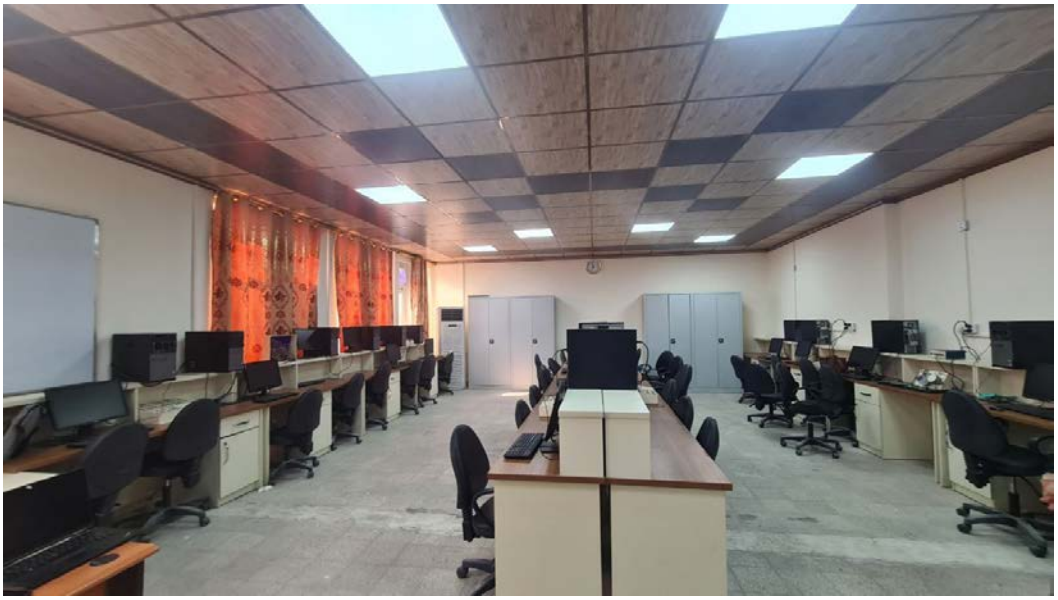
9. المراجع والمصادر:

- نايز، ن. س. هندسة أنظمة التحكم، الطبعة السابعة، وايلي.
- أوغاتا، ك. هندسة التحكم الحديثة، الطبعة الخامسة، برنتيس هول.
- مستندات MATLAB: https://www.mathworks.com/help/matlab/
- مستندات مجموعة أدوات تصميم التحكم LabVIEW: https://www.ni.com
- دليل مستخدم LabVIEW، ناشيونال إنسترومنتس.
- Nise, N. S. *Control Systems Engineering*, 7th Edition, Wiley.
- Ogata, K. *Modern Control Engineering*, 5th Edition, Prentice Hall.
- MATLAB Documentation: https://www.mathworks.com/help/matlab/
- LabVIEW Control Design Toolkit Documentation: https://www.ni.com
- LabVIEW User Manual, National Instruments.

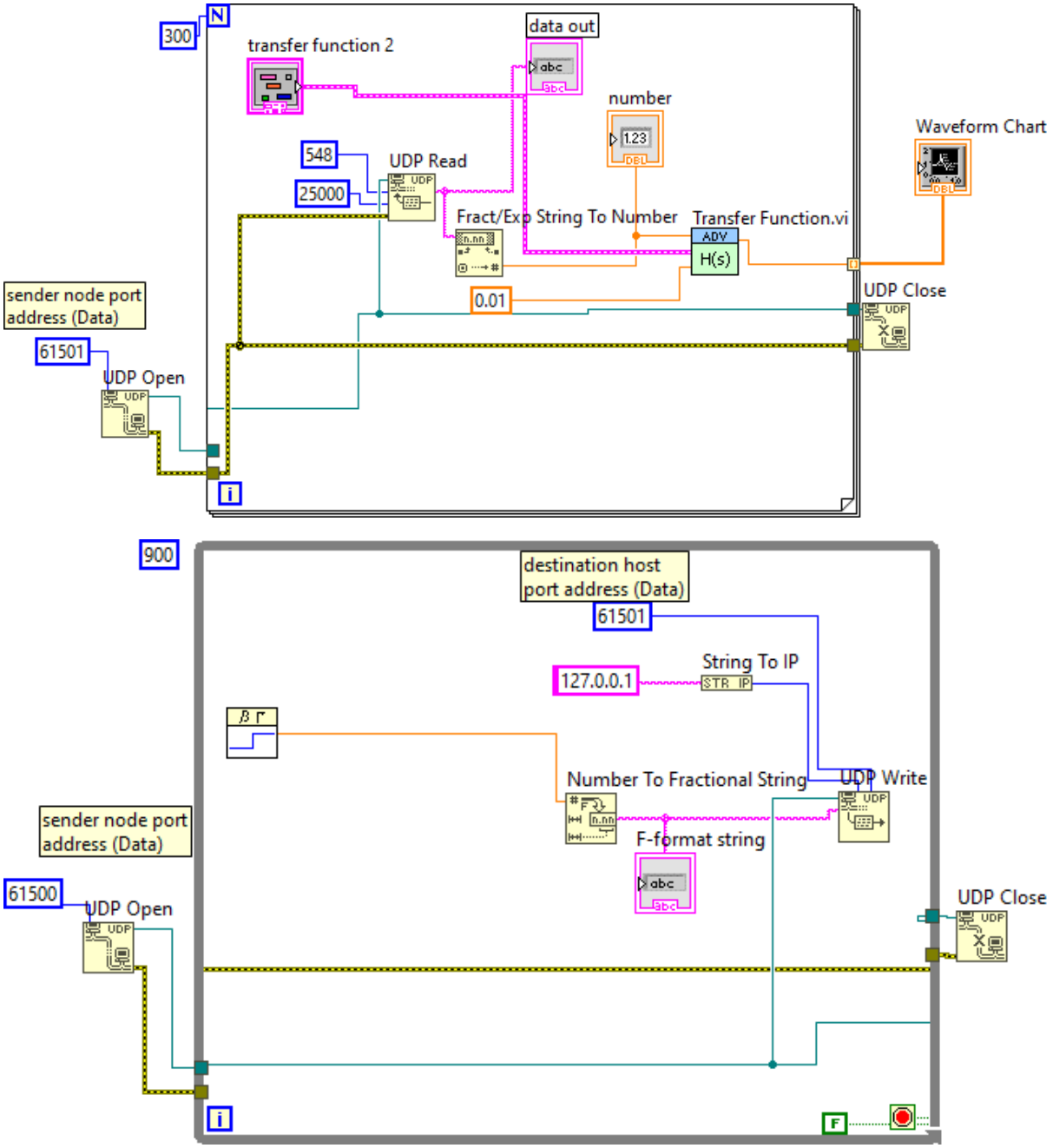
10.المرفقات:
• جدول المختبر: • شرح عن إجراءات التجربة(sheet) • نماذج ملفات LabVIEW VI • نماذج نصوص MATLAB:

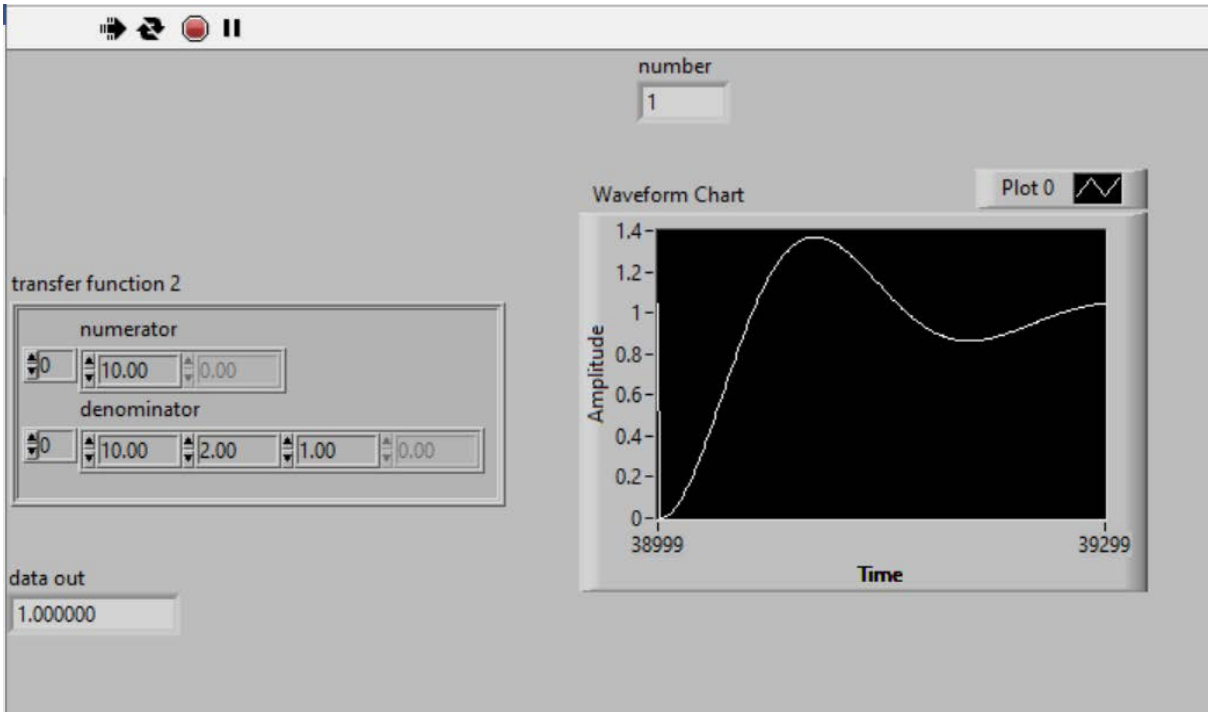
Laboratory tools instruments	
	Analogue Servo unit 110-33 , Mechanical unit control and instrumentation 100-33.
	Data Acquisition Cards.
	Raspberry Pi 4 and 3 versions. Arduino and sensors set.
	LOGO 12\24 REC PLC.

210
113LGE040011
مختبر السيطرة
113LGE04004
مختبر المعالجات
113LGE03003
مختبر المنطق



LabVIEW (.vi sample programs)




number
1
Waveform Chart
Plot 0
transfer function 2
numerator
0
10.00
0.00
denominator
0
10.00
2.00
1.00
0.00
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Amplitude
38999
39299
Time
data out
1.000000

M file sample (mathscript)

```
K1=2;K2=3;
s=tf('s');
G1=1/(s+1)/(s+2);
H1=s;
G2=1/(s+3)
H2=s^2+1
G3=(s^3+s)/((s+0.1)*(s+0.2))
H3=s+1;
F1=feedback (G1,H1)
F2=parallel (H2,G2)
F3=feedback (H3,G3)
F4=feedback(F2*F3,K1,+1)
F5=feedback(F4,K2/H3*F1)
F6=series (F5,1+F1)
```

University of Mosul
College of Engineering
Computer Engineering Dept.

Fourth Class
Control Lab.
Experiment:2

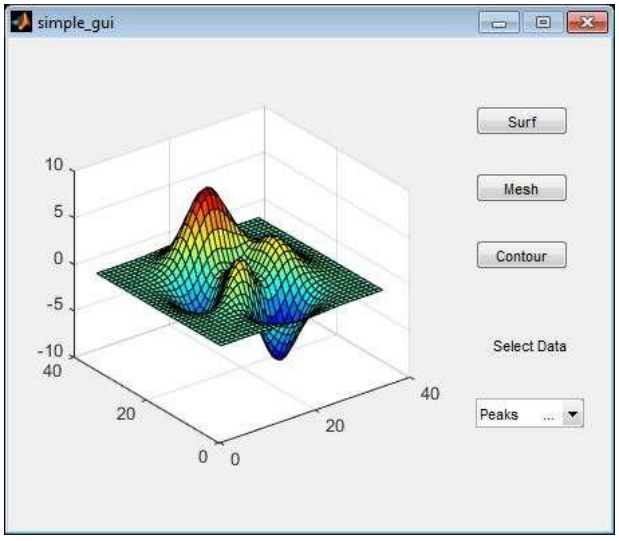
Graphical User Interface (GUI) in Matlab

What Is a UI?

A user interface (UI) is a graphical display in one or more windows containing controls, called components, that enable a user to perform interactive tasks. The user does not have to create a script or type commands at the command line to accomplish the tasks.

Unlike coding programs to accomplish tasks, the user does not need to understand the details of how the tasks are performed. UI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders—just to name a few. UIs created using MATLAB® tools can also perform any type of computation, read and write data files, communicate with other UIs, and display data as tables or as plots.

The following figure illustrates a simple UI that you can easily build yourself.



The UI contains these components:

• An axes component

• A pop-up menu listing three data sets that correspond to MATLAB functions: peaks, membrane, and sinc

• A static text component to label the pop-up menu

• Three buttons that provide different kinds of plots: surface, mesh, and contour.

When you click a push button, the axes component displays the selected data set using the specified type of 3-D plot.

Ways to Build MATLAB UIs:

A MATLAB UI is a figure window to which you add user-operated components. You can select, size, and position these components as you like. Using callbacks you can make the components do what you want when the user clicks or manipulates the components with keystrokes.

You can build MATLAB UIs in two ways:

• **Create the UI using GUIDE**

This approach starts with a figure that you populate with components from within a graphic layout editor. GUIDE creates an associated code file containing callbacks for the UI and its components. GUIDE saves both the figure (as a FIG-file) and the code file. You can launch your application from either file.

• **Create the UI programmatically**

Using this approach, you create a code file that defines all component properties and behaviors. When a user executes the file, it creates a figure, populates it with components, and handles user interactions. Typically, the figure is not saved between sessions because the code in the file creates a new one each time it runs.

The code files of the two approaches look different. Programmatic UI files are generally longer, because they explicitly define every property of the figure and its controls, as well as the callbacks. GUIDE UIs define most of the properties within the figure itself. They store the definitions in its FIG-file rather than in its code file. The code file contains callbacks and other functions that initialize the UI when it opens.

You can create a UI with GUIDE and then modify it programmatically. However, you cannot create a UI programmatically and then modify it with GUIDE.

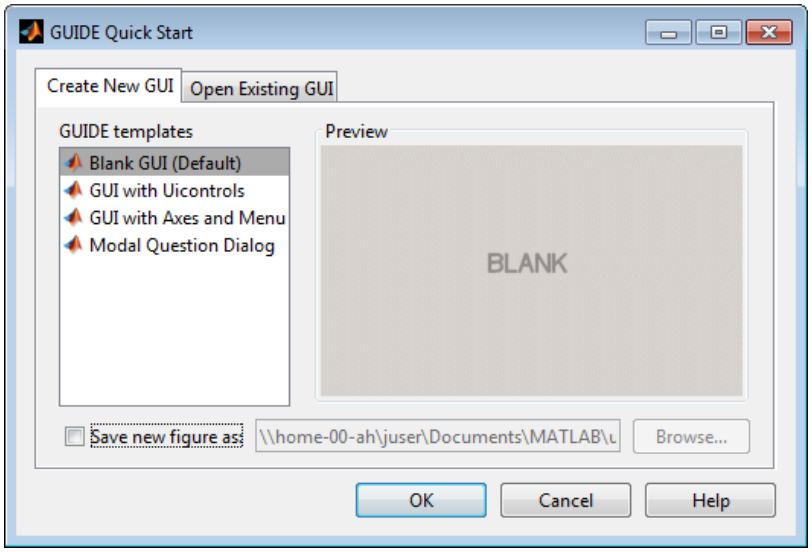
How to Create a UI with GUIDE?

We are going to develop a simple **Matlab GUI**. We'll use the **Matlab GUIDE** (**Graphical User Interface Development Environment**) which is pretty handy.

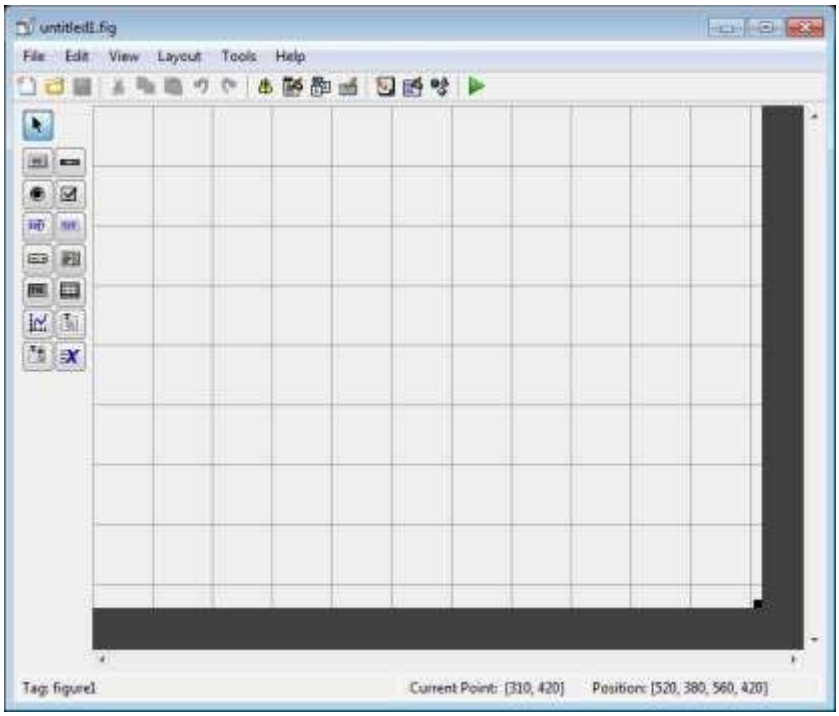
Let's start the ride! On the command window type:

>>guide

This will open the '**Quick Start**' window, where you can study or review their examples, too! Follow the steps and select the first option: '**Blank GUI (Default)**'.



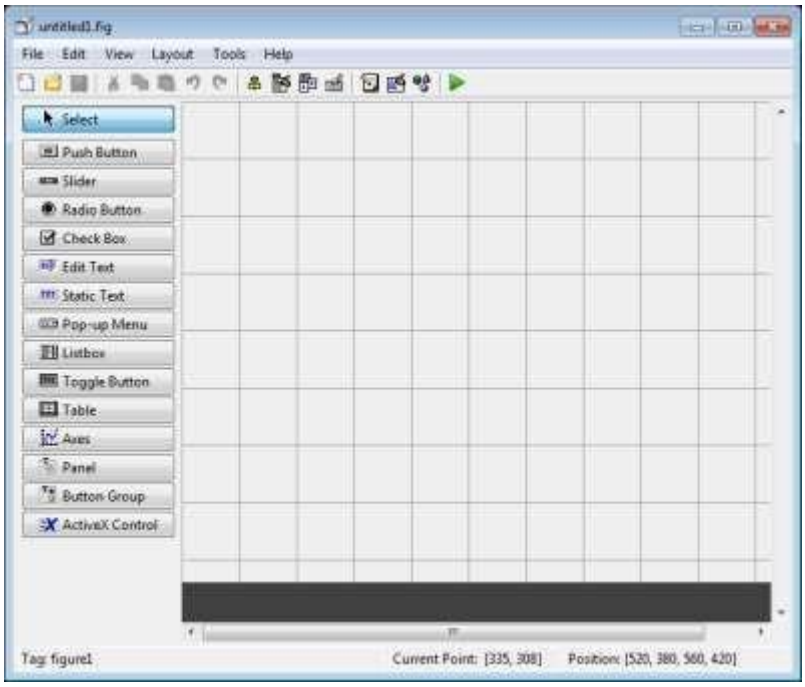
Then, an untitled figure will pop-up. You have some components on the left menu, which you can drag onto your interface.



Display the names of the UI components in the component palette:

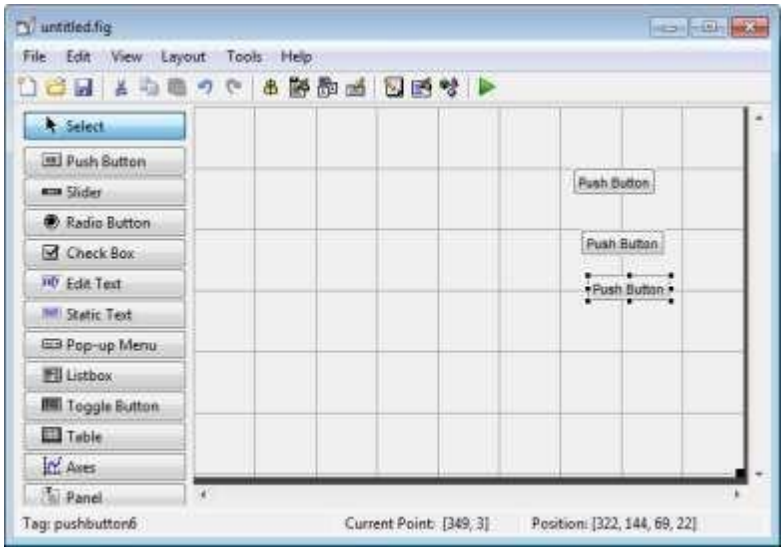
1. Select File > Preferences > GUIDE.
2. Select Show names in component palette.

3. Click OK.



Add, align, and label the components in the UI:

Add push buttons to the UI. Select the push button tool from the component palette at the left side of the Layout Editor and drag it into the layout area. Create three buttons, positioning them approximately as shown in the following figure.



Align the Components:

If several components have the same parent, you can use the Alignment Tool to align them to one another. To align the three push buttons:

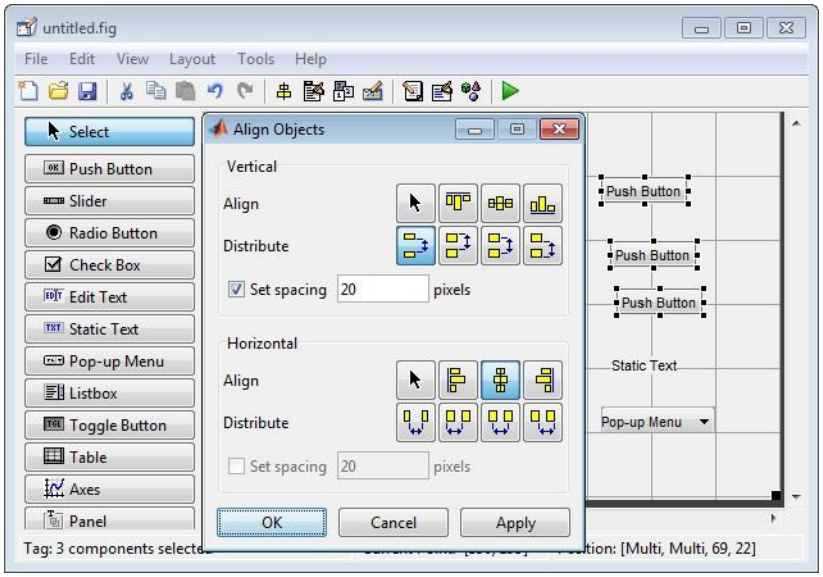
1 Select all three push buttons by pressing Ctrl and clicking them.

2 Select Tools > Align Objects.

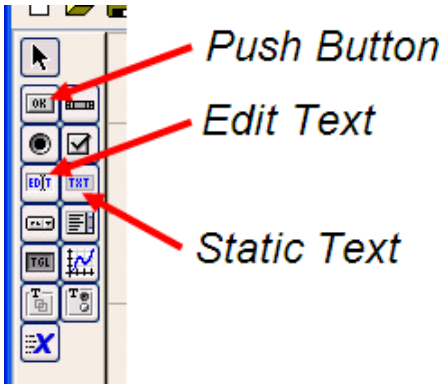
3 Make these settings in the Alignment Tool:

• Left-aligned in the horizontal direction.

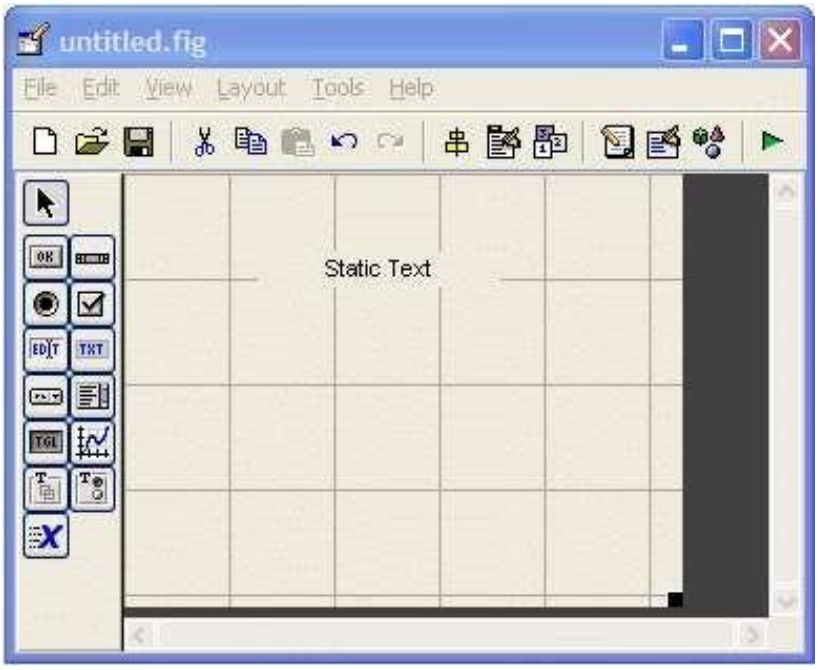
• 20 pixels spacing between push buttons in the vertical direction.



In this example we are going to use only two '**push buttons**' and one '**static text**'.



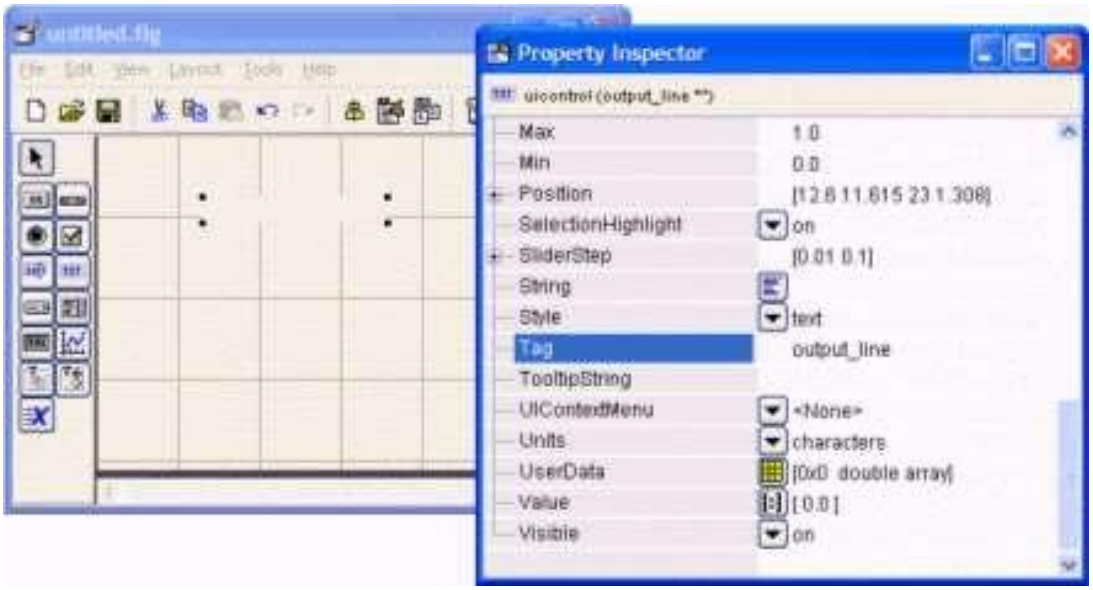
Drag and drop a '**static text**' onto your Matlab GUI. You can reduce or increase the size of your interface window by dragging its bottom-right corner, as it's done in other drawing programs.



Double click on this '**static text**' and a '**Property Inspector**' window will appear. Scroll down and look for the '**String**' property and delete what's in there. For the moment we want it to be blank.

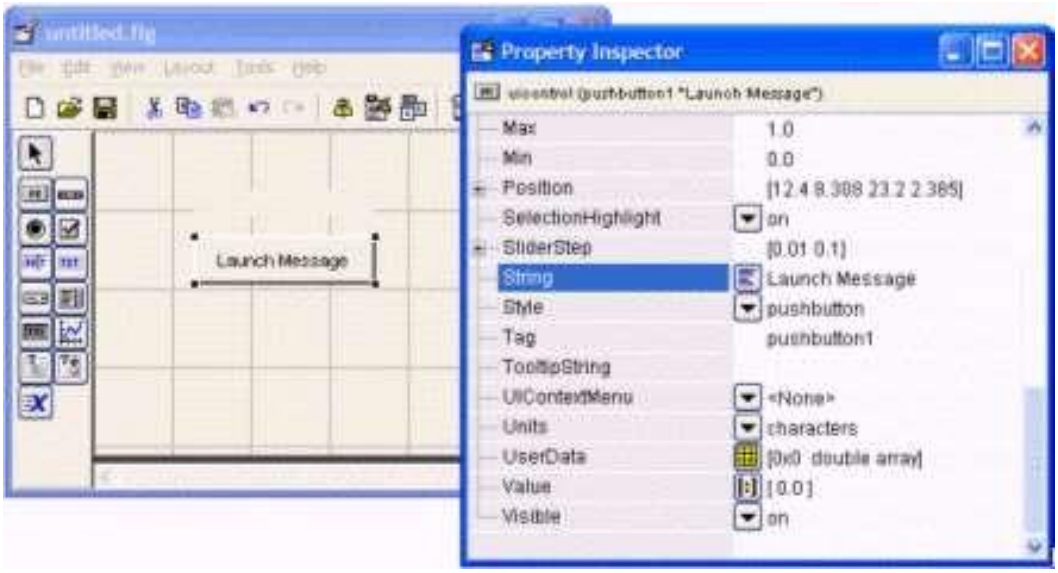
Then, make the '**Tag**' property to be '**output_line**'. You can use whatever name you want.

Your windows must look similar to the figure below:



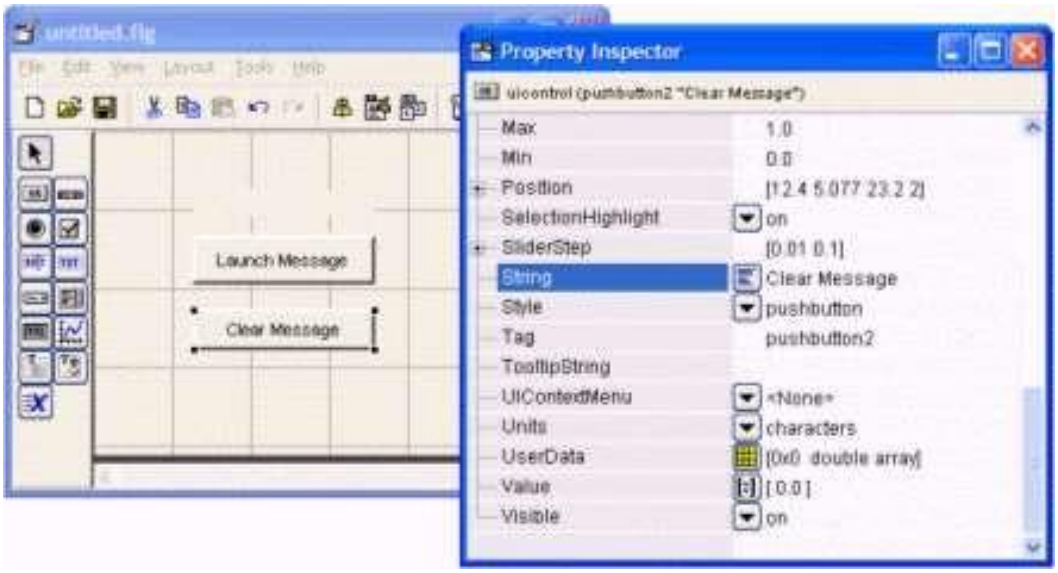
Then, drag-and-drop a '**push button**' onto your interface. Modify its '**String**' property to read '**Launch Message**'. Let its '**Tag**' property intact. You could change this tag... it's the name or identifier of the object as it's going to be recognized in the rest of the code.

Your windows must look similar to the figure below:

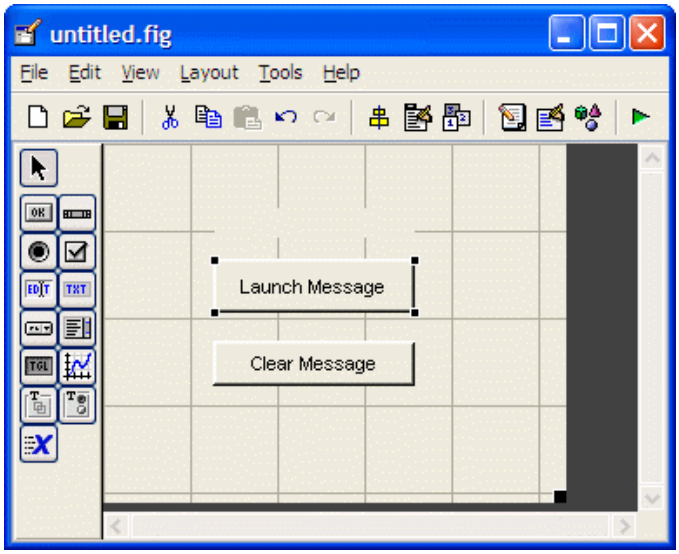


Drag-and-drop another '**push button**'. Modify its '**String**' property to read

'**Clear Message**' and leave its '**Tag**' as it is. You'll produce these results.



Now, right-click on the '**Launch Message**' button and choose '**View Callbacks**' -> '**Callback**'



You'll be asked to save your figure. A good name is **hello_world.fig**... use the name that you like.

You'll be taken to the Matlab code (in the editor window) that will drive your interface. Matlab has **automatically** created functions related to your

components. You have to make the final touches... For the moment, don't worry about the many lines automatically created. Just focus on what we need to do.

The '**Callback**' functions are the instructions that will be executed when the user pushes the buttons or does something with the components that you have included in your Matlab GUI. In this case, you'll see something like this code.

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MAT
% handles    structure with handles and user data (see GUIDATA)
```

A '**set**' instruction sets the properties of the elements that you indicate. Do you remember that you have a '**static text**' with the tag (identifier) '**output_line**'? We are going to modify it when the user pushes the button with the string (or label) '**Launch Message**'. This is accomplished with the instruction:

```
set(handles.output_line,'String','Hello  World!!')
```

The **first parameter is the object** (component) that you're going to modify. It starts with '**handles.**'. The **second argument is the object's property** that you're going to modify, and in this case is the '**String**' property. The **third argument is the value** that you want to assign to the property.
So, the result is that when the user presses the '**Launch Message**' button, a message reading '**Hello World!!**' will appear in the '**output line**' (officialy named '**handles.output_line**'). Add this single line to the code, so that it

looks like this:

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MAT
% handles    structure with handles and user data (see GUIDATA)
set(handles.output_line,'String','Hello World!!')
```

We'll do something similar to the '**callback**' corresponding to the '**Clear Message**' button. So change this original code...

```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MAT
% handles    structure with handles and user data (see GUIDATA)
```

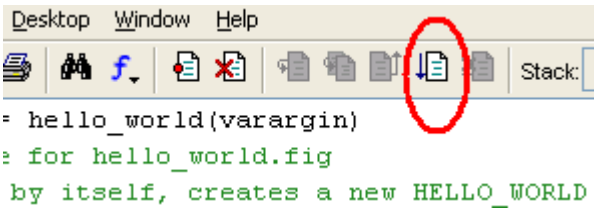
into this...

```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MAT
% handles    structure with handles and user data (see GUIDATA)
set(handles.output_line,'String','')
```

The result is that when the user presses the '**Clear Message**' button, a blank message will appear in the '**output line**' (officialy named '**handles.output_line**').

Magic is about to happen!

Now, run your interface by clicking the '**run**' icon at the top of the editor window...



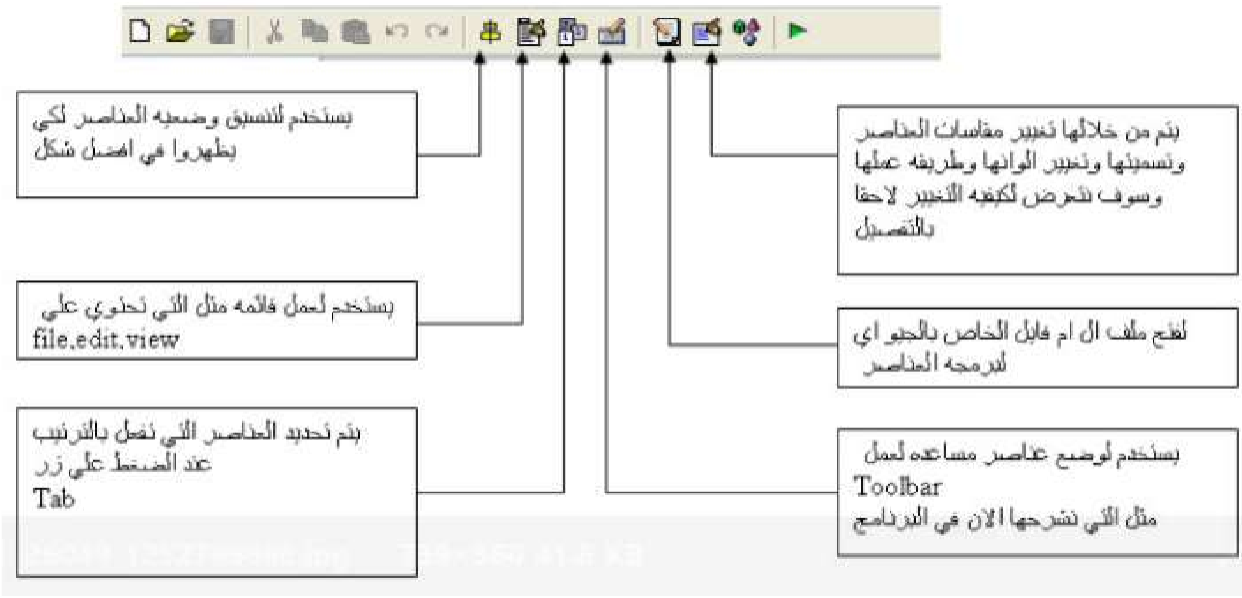
Try, it! Press the '**Launch Message**' button... and an interesting message appears...

then, press the '**Clear Message**' button...

let's summarize:

- You can drag-and-drop your components onto your graphic interface to start your Matlab GUI.
- Matlab will automatically create callback-functions, related to the buttons or other components that you include.
- The 'set' instruction assigns values to properties of the elements that you want to modify. The general format is:

set(handles.tag_of_your_component, 'Property', value)



التطبيق الاول:

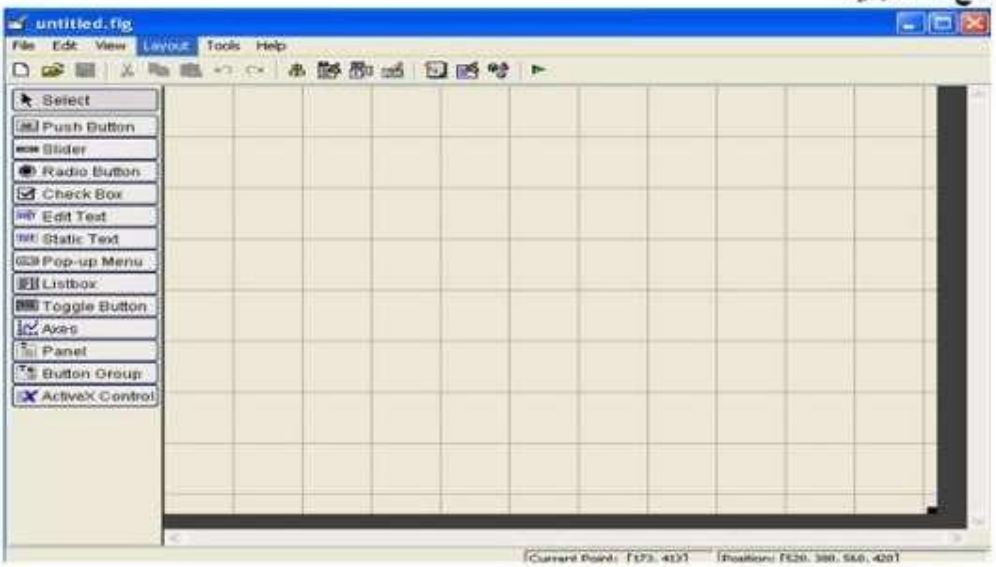
ادخال رقمين ثم اختيار العمليه المراد تطبيقها ثم اظهار الناتج

الغرض: التعرف علي برمجه ال

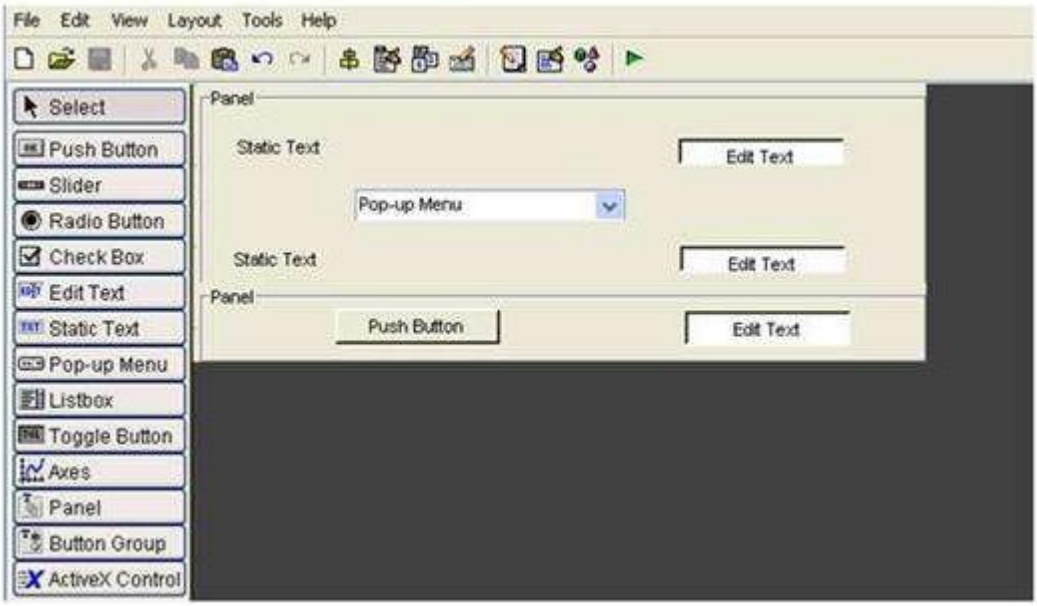
1-push button
2-pop-up menu

الخطوات:

فتح ملف جديد



سحب وادراج العناصر علي الخلفيه كما يظهر في الصوره ,ثم تصغير الخلفيه لتلائم التصميم



انوقوف علي كل عنصر منفصلا وانضغط علي زر الماوس الايمن واختيار

Property inspector

وانتزول من الجدول حتي الوصول الي هذه الخانه

ثم تفريغ الخانه من الكلام في الثلاثه عناصر ل

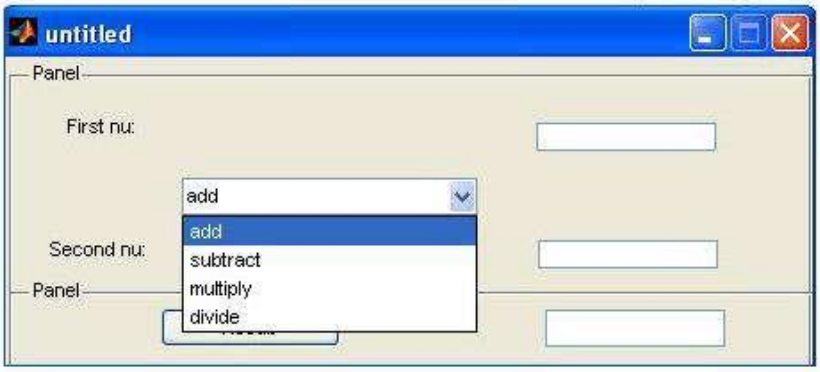
Edit text

اما باقي العناصر فيكتب الاسم المراد ظهوره للمستخدم
في حالة ل

Pop-up menu

فنضغط علي هذا الرمز ونكتب ما يلي

فيصبح البرنامج بهذا الشكل عند تشغيله

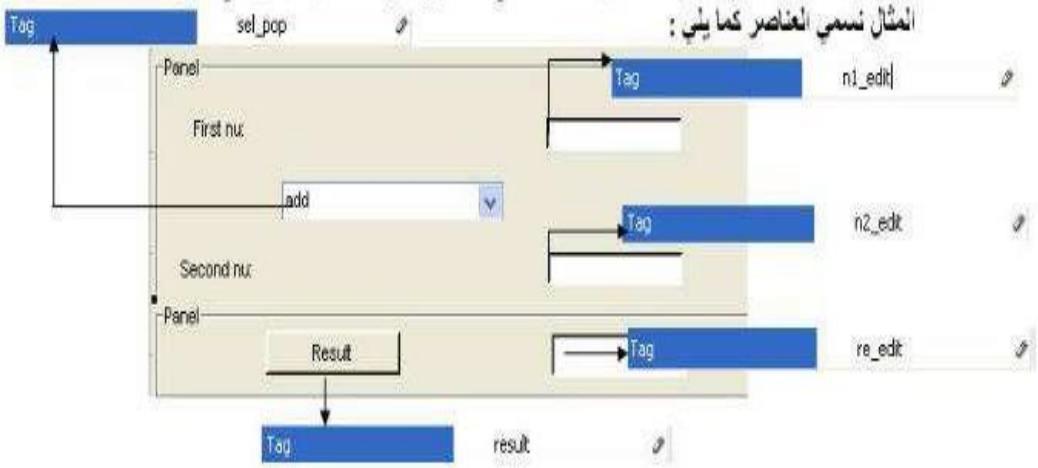


بالرجوع الي خلفيه التصميم والضغط علي الزر الايمن للماوس واختيار

Property inspector

والنزول في الجدول الي ان نصل الي هذه الخانه

من خلال هذه الخانه يتم تسميه العناصر بالاسماء التي نستدعيها بها اثناء البرمجه و علي سبيل المثال نسمي العناصر كما يلي :



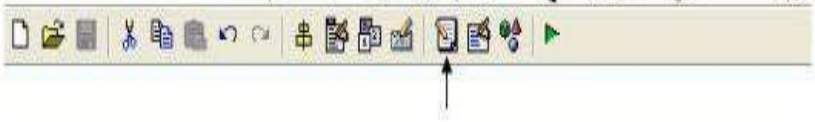
جميع الخطوات السابقه ادت الي وصولنا الي التصميم النهائي والمتبقي فقط هو برمجه هذا التصميم
لذلك علينا فهم اسلوب عمل البرنامج قبل بدايه البرمجه
عند الضغط علي

Result

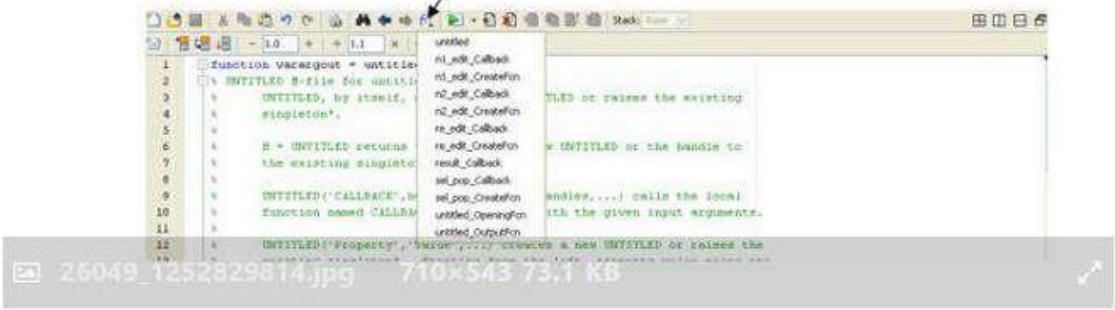
يتم جلب الكتابه المكتوبه في المكان الاول وتحويلها الي ارقام ليستطيع التعامل معها
وكذلك الكتابه المكتوبه في المكان الثاني وتحويلها الي ارقام ايضا
وبعد ذلك يري رقم الخيار الذي اختاره المستخدم للعمليه الحسابيه
فالرقم واحد يعني جمع
والرقم اثنان يعني الطرح
وثلاثه الضرب
واربعه تعني القسمه

الان علينا تنفيذ ما فهمناه

نقوم بالضغط علي هذا الرمز لفتح ملف ال ام فايل المرتبط بالتصميم



ثم الضغط علي هذا الرمز داخل أل ام فايل لنري جميع العناصر



نختار الاسم هذا من القائمه للزهاب الي زر

Result

عند اختيار هذا الاسم نجد ان الساوس وقف عند هذا الجزء من الكود

```
function result_Callback(hObject, eventdata, handles)
% hObject    handle to result (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

تحت الكتابه المكتوبه باللون الاخضر يتم كتابه كود برمجه الزر
وهي كما يلي

```
n1=str2num(get(handles.n1_edit,'string'));
n2=str2num(get(handles.n2_edit,'string'));
sel=get(handles.sel_pop,'value');
switch sel
    case 1
        c=n1+n2;
    case 2
        c=n1-n2;
    case 3
        c=n1*n2;
    case 4
        c=n1/n2;
end
```

فتظهر النتيجة بالشكل النهائي: