



US 20180178380A1

(19) **United States**

(12) **Patent Application Publication**
Oestergaard et al.

(10) **Pub. No.: US 2018/0178380 A1**

(43) **Pub. Date: Jun. 28, 2018**

(54) **METHOD FOR EXTENDING END USER PROGRAMMING OF AN INDUSTRIAL ROBOT WITH THIRD PARTY CONTRIBUTIONS**

Publication Classification

(51) **Int. Cl.**
B25J 9/16 (2006.01)
(52) **U.S. Cl.**
CPC *B25J 9/1664* (2013.01); *B25J 9/161* (2013.01); *G05B 2219/39548* (2013.01); *G05B 2219/40099* (2013.01); *Y10S 901/03* (2013.01)

(71) Applicant: **Universal Robots A/S**, Odense S. (DK)

(72) Inventors: **Esben H. Oestergaard**, Odense S. (DK); **Martin Slota**, Odense S. (DK); **Robert Wilterdink**, Odense S. (DK)

(57) **ABSTRACT**

There is provided a method for programming an industrial robot, where distributors and integrators can present accessories that run successfully at end users. Also the developer can define customized installation screens and program nodes for the end user. There is provided a software platform, where the developer can define customized installation screens and program nodes for the end user thereby extending an existing robot system with customized functionalities by still using the software platform available in the robot system. Hereby a robot developer can define customized installation screens and program nodes for the end user. These can, for example, encapsulate complex new robot programming concepts, or provide friendly hardware configuration interfaces.

(21) Appl. No.: **15/738,315**

(22) PCT Filed: **Jul. 7, 2016**

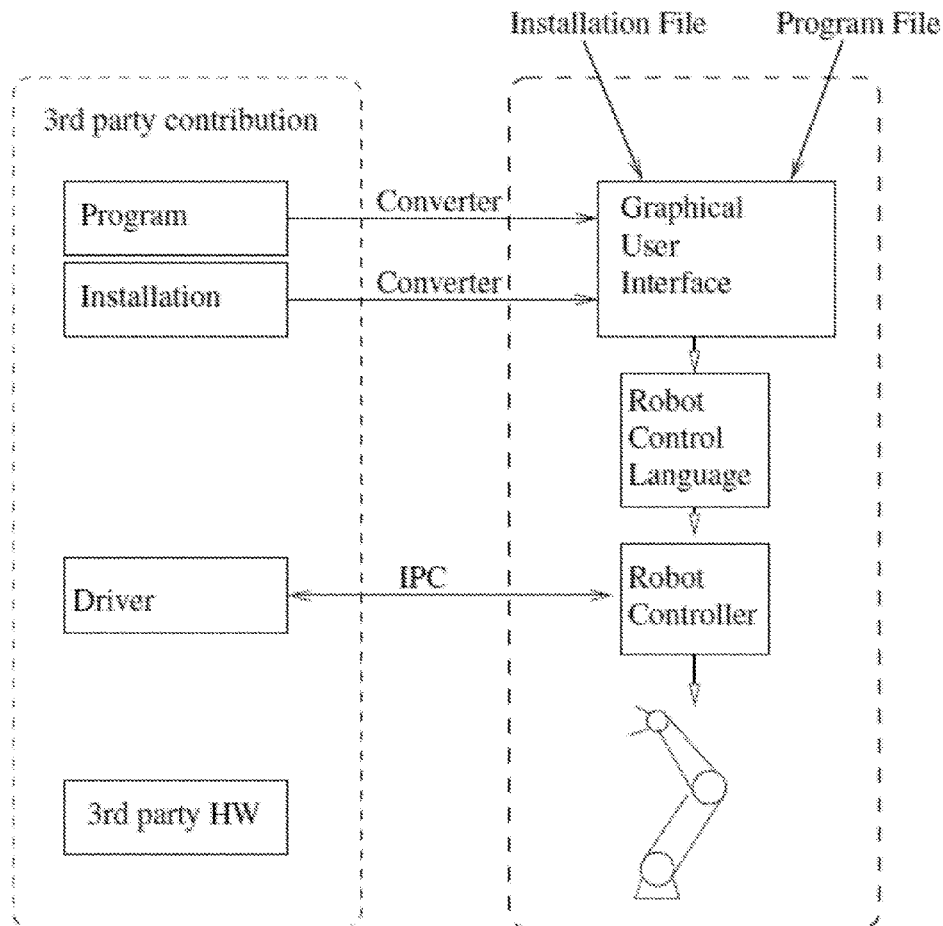
(86) PCT No.: **PCT/DK2016/050240**

§ 371 (c)(1),

(2) Date: **Dec. 20, 2017**

(30) **Foreign Application Priority Data**

Jul. 8, 2015 (DK) PA 2015 70451



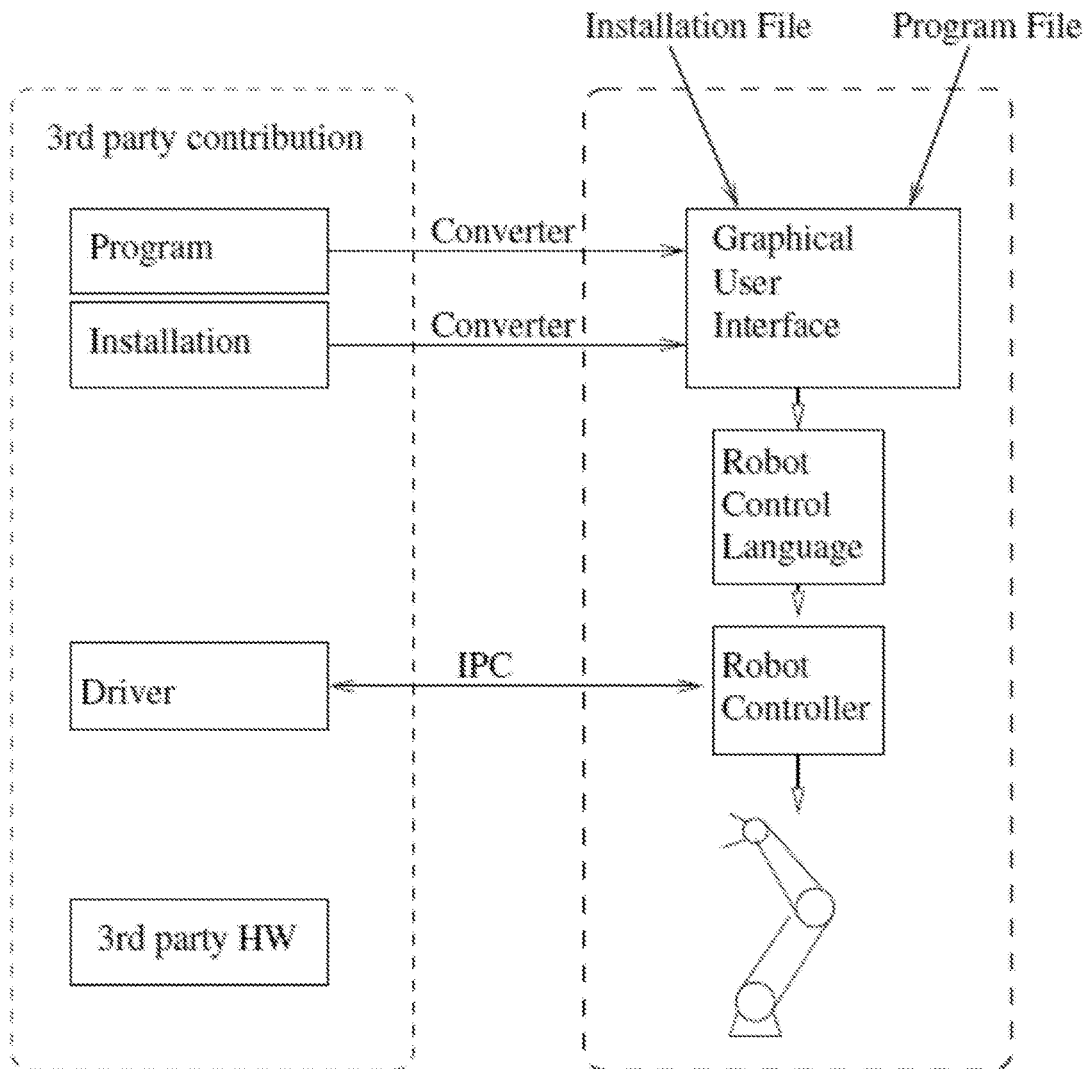


Figure 1

**METHOD FOR EXTENDING END USER
PROGRAMMING OF AN INDUSTRIAL
ROBOT WITH THIRD PARTY
CONTRIBUTIONS**

FIELD OF THE INVENTION

[0001] The present invention relates to a method for programming an industrial robot. More specifically, the invention relates to a method for programming an industrial robot, where distributors and integrators can present accessories that run successfully at end users. Also the developer can define customized installation screens and program nodes for the end user. The invention provides a software platform, where the developer can define customized installation screens and program nodes for the end user.

BACKGROUND OF THE INVENTION

[0002] Before an industrial robot is to be put in operation for a certain task, it must be programmed to carry out the task. Various robot programming languages are available originating from different robot suppliers. A robot program comprises a series of robot instructions written in a robot programming language.

[0003] Customers who buy industrial robots generally want to be able to control or manipulate a robot, and to program the robot, relative to various objects and boundaries in the surroundings of the robot, such as machines, objects or blanks, fixtures, conveyers, pallets or vision systems.

[0004] The end-user would sometimes face a challenge when installing and programming third party hardware (tools, such as grippers) as well as accompanying third party software to be integrated in the industrial robot system.

[0005] Industrial robots are highly flexible devices used for a wide variety of operations in many different industrial applications. Industrial robots are conventionally programmed via a robot programming language that is very similar to conventional computer programming languages. A robot program includes a sequence of program instructions, where each instruction tells the robot control unit what to do and how to do it. Robots are programmed to follow a path including a plurality of target points. The robot program includes the positions of the target points. The programming of robots is a time consuming process and the conventional methods of using the robot during the programming and teaching process ties up the production equipment and delays production start.

[0006] There is a need for a method to extend an existing robot system with customized functionalities by still using the software platform available in the robot system. Hereby a robot developer can define customized installation screens and program nodes for the end user. These can, for example, encapsulate complex new robot programming concepts, or provide friendly hardware configuration interfaces.

SUMMARY OF THE INVENTION

[0007] The above and other objects and advantages are obtained by the provision of a programmable robot having the unique feature that it can be programmed with third party plug-ins provided with the third party tools.

[0008] Thus, according to a first aspect of the present invention there is provided a method of programming an industrial robot with third party hardware and software, wherein the method includes:

[0009] providing an industrial robot having a user graphical interface (GUI) for programming the robot and installing third party software, said GUI including i) an installation tab, wherein the installation of the robot is specified, said installation tab configured to allow setting up of parameters for the third party hardware mounted on the robot, and ii) a program tab, wherein the program of the robot is specified, said program tab configured to allow setting up of the steps to be performed by the hardware;

[0010] installing the third-party software, based on a dedicated file format, that adds a GUI element to the installation tab and the program tab;

[0011] installing third-party hardware, such as a tool, based on said software components containing GUI elements.

[0012] The method may further comprise one or more of the following steps:

[0013] storing in a memory location a set of predefined positions comprising preprogrammed robot code to define one or more geometrical features relative to objects in the surroundings of the robot and establishing a relationship between said geometrical features and first coordinates of a robot-related coordinate system;

[0014] integrating the GUI on a graphical display device, displaying information about the predefined positions and allowing a user to select one or more of the predefined positions and to specify the order in which the robot shall visit the selected positions,

[0015] receiving information about which of the predefined positions the user has selected and the order in which the robot shall visit the selected workstations, and generating a robot program for performing a work cycle based on said predefined positions and said received and stored information about the selected positions and the order in which the robot shall visit the selected positions.

[0016] According to a further aspect of the invention, the object is achieved by a computer program directly loadable into the internal memory of a computer or a processor, comprising software code portions for performing the steps of the method according to the invention, when said program is run on a computer. The computer program is provided either on a computer readable medium or through a network.

[0017] According to another aspect of the invention, the object is achieved by a computer readable medium having a program recorded thereon, when the program is to make a computer perform the steps of the method according to the invention, and said program is run on the computer.

[0018] According to the invention, a set of predefined workstations, comprising preprogrammed robot code, are stored in a memory location. The user creates a robot program by selecting one or more workstations from the predefined set of workstations and specifying the order in which the robot shall visit the selected workstations. This allows the operator to create a graphical representation that reflects the configuration of the real robot cell.

[0019] A workstation is a physical location on which the robot may carry out work. Each predefined workstation represents a real workstation and a defined task to be carried out by the robot at the workstation. The preprogrammed robot code is specially designed for controlling the robot so that it carries out the defined task at the workstation.

Examples of workstations are: an injection mold machine, a dye casting machine, a scrap station for throwing away rejected objects, a device for vision control of the object, one or more stations for subsequent treatment of the object, and an output station such as a conveyer. Examples of tasks are: pick a molded object from the mold, throw the object in the waste basket, hold the object during a vision control, move the object in a predefined way in relation to a treatment tool, put the object in an assembly with other objects, place the object on the conveyer. The subsequent treatment is for instance flaming, deburring, degating, dispensing, polishing, grinding or painting.

[0020] The user interacts with one or more graphical interfaces during the creation of the robot program. In this context a graphical interface is defined as a view or a part of a view, adapted for interaction with a user and displayed on the display device. The graphical interface is adapted for displaying information, in the form of text and symbols, and choices to the user, and to receive commands, data and selections entered by the user.

[0021] A robot program is generated based on the stored, predefined workstations including predefined robot program code, and information entered by the user via the user interfaces about selected workstations and the order in which the robot shall visit the selected workstations. The programming code is hidden from the user, and the user creates a robot program by interacting with one or more graphical interfaces. Thus, the invention makes it possible for a user to create a robot program without using any programming code. The invention offers an easy and pedagogical way to program a robot, and does not require any knowledge about robot languages. Based on the software platform provided with the present invention it is easy for the user to reprogram the robot in connection with adding and removing a third party workstation to the robot cell. A further advantage gained with the present invention is that the user will find it just as easy to program an advanced robot with five or six axes, as to program a simple linear robot with only three axes. The term industrial robot refers to linearly movable manipulators as well as robots including rotational movement axes. The invention is useful for programming industrial manipulators or robots for any application including visiting and performing work on one or more workstations.

BRIEF DESCRIPTION OF THE DRAWING

[0022] FIG. 1 shows a diagram of the relationship between the software and hardware components supplied with the robot system and the third-party supplied additions to this system.

DETAILED DESCRIPTION OF THE INVENTION

[0023] The invention will now be explained more closely by the description of the principles shown in FIG. 1. Specifically the FIGURE shows how the programming of an industrial robot with third party hardware and software is carried out. An industrial robot having a user graphical interface (GUI) for programming the robot is provided. The installation of the third party software is performed where the GUI includes an installation tab, wherein the installation of the robot is specified, said installation tab configured to allow setting up of parameters for the third party hardware

mounted on the robot, and a program tab, wherein the program of the robot is specified, said program tab configured to allow setting up of the steps to be performed by the hardware. Then the third-party software is installed, based on a dedicated file format that adds a GUI element to the installation tab and the program tab. Finally the third-party hardware (or own hardware) is installed based on said software components containing GUI elements.

[0024] The invention concerns a platform where distributors and integrators can present accessories that run successfully at end users. This gives rise to shorter lead time to implement robot applications, and lower project risks, higher predictability of project costs and time, access to well-proven technology, and share information. The concept includes both hardware and software solutions, whereas the present invention is concerned with the software aspects of this concept.

[0025] By implementation of the present invention an industrial robot becomes receptive to the installation of third-party software components, based on a dedicated file format. These files can amongst other things contain graphical user interface elements that will become part of our user interface. A typical use case could be the installation of a gripper on the robot. The gripper would come with a dedicated file, which would add a general user interface element to the installation tab (where the installation of the robot is specified) and to the program tab (where the program of the robot is specified). In the installation tab, it would allow the set-up of parameters for how the gripper is mounted on the robot and what electrical signals are used for the gripper. In the program tab, it would allow to specify whether the gripper should open or close, and possible also the gripping force, closing speed etc.

[0026] Generally, an industrial robot comprises a manipulator, a control unit for controlling the manipulator, and a portable operator control device, denoted a teach pendant unit, for teaching and manually operating the manipulator.

[0027] In accordance with the present invention end-users have the possibility to install various plugin packages. A single package may contain a contribution to the installation tab and/or program nodes. After installation of the software the new screens will automatically become available for the control device. Furthermore, the screens look and act as if they were part of the default control environment.

[0028] Installation tab contributions store their settings in the, so called, installation. This file contains the configuration for a particular work-cell. The configuration among others contains the verified safety settings. End-users programming robots, using the present invention, automatically link the program under design to the specific installation. Consequently, when the program is loaded, the assigned installation (including verified safety configuration and settings) is loaded as well. This simplifies the workflow for end-users, because the number of significant configuration files remains the same (i.e. one).

[0029] Program node contributions store their parameters in the, so called, program files. These files contain the program for the automation task the robot is supposed solve. The combination of default control program parameters and specific parameters gives the same workflow advantages to the end-user (similar to the installation file), since the number of significant files remains one.

[0030] The separation of concerns, between configuration of the robot work-cell and program design, promoted by the

control device lowers the design complexity for end-users. Both concerns focus on complementary aspects of the automation task and give the end-user a systematic way of solving the automation problem. The control device ensures that both concerns are integrated in an appropriate manner. With the framework of the present invention the same separation of concerns and integration can be provided to end-users, i.e. installation tab contributions can share settings with program node contributions.

[0031] For instance, if an electro-mechanical gripper is supposed to be installed and used for an automation task the following steps could be taken. As the gripper supplier chose to guide the end-user through the installation process, he directs the end-user to install the plug-in first and to navigate to the Installation tab. In the Installation tab the contribution then shows the end-user visually how the gripper's power supply and communication interface should be connected. After the communication interface and power have been connected physically, the gripper boots up and starts communicating with the gripper software. At this point, the gripper software detects the physical gripper and the screen changes from showing the visual installation guide to a configuration interface. In the configuration interface, for instance, the maximum closing force can be configured. After the configuration has been done, the end-user can start making a program. The end-user navigates to the program tab and starts programming his task. At the point where he wants to fetch an object he looks for an appropriate program node and finds the "Gripper" node. He inserts the gripper node in the same manner as any other program node. The command tab (for program node parametrization) shows the available parameters that can be tuned in a visually appealing manner. After the end-user specified the gripper opening distance and closing distance, the gripper node is parameterized. To test the gripper parameters, the command tab offers a "Joggle" button which will open and close the gripper, without running the entire program. Before any program can be run, each program node will need to generate a code that executes the specified actions. Information stored in the installation, e.g. the maximum closing force, can be employed here as well. For the gripper node, the developer has taken care of the script generation (using the framework of the present invention). On pressing the play button in the control device, the program will be converted into the script of the robot and the robot will start performing the programmed task.

[0032] The framework of the present invention enables:

[0033] Easy setup of third party hardware, which is usually a complex task not suited for regular end-users (like factory workers). The benefits are: lower technical expertise required, less time for setup and allows easy repetition for setup of 3rd party hardware->lower setup costs.

[0034] Seamless integration with the normal workflow.

[0035] Both the framework and control device promote the same workflow.

1. A method of programming an industrial robot, the method comprising includes:

displaying a graphical user interface (GUI) for programming the industrial robot and for installing third-party software on a controller associated with the industrial robot, the GUI comprising:

an installation tab specifying an installation of the industrial robot, the installation tab being configured

for specifying parameters for hardware, including third-party hardware, on the industrial robot, and a program tab specifying programming of the industrial robot, the program tab being configured for specifying steps to be performed by the hardware; and installing the third-party software on the controller associated with the industrial robot, the third-party software being based on a dedicated file format and adding a GUI element to at least one of the installation tab or the program tab;

wherein the third-party software and the GUI element enable installation of the third-party hardware on the industrial robot.

2. The method of claim 1, further comprising:

storing, in memory, predefined positions represented by preprogrammed robot code to define one or more geometrical features relative to objects in surroundings of the industrial robot; and

establishing a relationship between the geometrical features and first coordinates of a robot-related coordinate system.

3. The method of claim 2, wherein the GUI is displayed on a graphical display device; and

wherein the method further comprises:

displaying, on the graphical display device, information about the predefined positions such that one or more of the predefined positions is selectable by a user, and such that an order in which the industrial robot moves to selected ones of the predefined positions is specifiable by the user.

4. The method of claim 3, further comprising:

receiving information about one or more of the predefined positions that the user has selected and the order in which the industrial robot is to move to the one or more of the predefined positions that the user has selected; and

generating a robot program for performing a work cycle based on the information about the one or more predefined positions that the user has selected and the order in which the industrial robot is to move to the one or more of the predefined positions that the user has selected.

5. The method of claim 1, wherein the industrial robot has an extensible operating system for allowing third-party daemons, servers, or computer programs to be installed on the controller to extend a functionality of the controller.

6. The method of claim 1, wherein the third-party hardware comprises an electro-mechanical gripper, and wherein the method further comprises:

receiving, by the controller, selection of the GUI element in the installation tab;

displaying, on a graphical display device, information about how a power supply of the mechanical gripper and a communication interface of the mechanical gripper should be connected;

the mechanical gripper booting-up and starting communicating with the GUI element thereby enabling detection of the of the mechanical gripper by the third-party software;

displaying, on the graphical display device, a configuration interface enabling configuration of one or more physical parameters relating to the mechanical gripper, including a maximum closing force of the mechanical gripper; and

receiving, via the program tab, programming for a task to be performed by the industrial robot using the mechanical gripper.

7. The method of claim 1, wherein the third-party hardware comprises a vision system, and wherein the method further comprises:

receiving, by the controller, selection of the GUI element in the installation tab;

displaying, on a graphical display device, information about how the vision system should be connected to the industrial robot, information about how coordinates of the vision system relate to coordinates of the industrial robot, and information about one or more parts to identify based on the vision system, the program tab comprising a GUI element enabling programming the industrial robot to move to an identified part; and

generating a robot program for performing a work cycle by moving the identified part to a predefined position, the robot program for performing a mathematical transformation between the coordinates of the vision system and the coordinates of the industrial robot for the identified part.

8. The method of claim 1, wherein the third-party hardware comprises an interface to industrial equipment, and wherein the method further comprises:

displaying, on a graphical display device, information in the installation tab about connecting the industrial robot to the industrial equipment; and

incorporating functions into the programming tab to include triggering events or functions in the industrial equipment.

9. A robot comprising:

a display device to display a graphical user interface (GUI) for programming the robot and for installing third-party software on a controller associated with the robot, the GUI comprising:

an installation tab specifying an installation of the robot, the installation tab being configured for specifying parameters for hardware, including third-party hardware, on the robot, and

a program tab specifying programming of the robot, the program tab being configured for specifying steps to be performed by the hardware; and

a controller configured to execute third-party software, the third-party software being based on a dedicated file format and adding a GUI element to at least one of the installation tab or the program tab;

wherein the third-party software and the GUI element enable installation of the third-party hardware on the robot.

10. The robot of claim 9, further comprising:

memory storing predefined positions represented by pre-programmed robot code to define one or more geometrical features relative to objects in surroundings of the robot;

wherein the controller is configured to establish a relationship between the geometrical features and first coordinates of a robot-related coordinate system.

11. The robot of claim 10, wherein the display device is to display information about the predefined positions such that one or more of the predefined positions is selectable by a user, and such that an order in which the robot moves to selected ones of the predefined positions is specifiable by the user.

12. The robot of claim 11, wherein the controller is configured to perform operations comprising:

receiving information about one or more of the predefined positions that the user has selected and the order in which the robot is to move to the one or more of the predefined positions that the user has selected; and

generating a robot program for performing a work cycle based on the information about the one or more predefined positions that the user has selected and the order in which the robot is to move to the one or more of the predefined positions that the user has selected.

13. The robot of claim 9, wherein the robot has an extensible operating system for allowing third-party daemons, servers, or computer programs to be installed on the controller to extend a functionality of the controller.

14. The robot of claim 9, wherein the third-party hardware comprises an electro-mechanical gripper;

wherein the controller is configured to receive data representing selection of the GUI element in the installation tab;

wherein the display device is configured to display information about how a power supply of the mechanical gripper and a communication interface of the mechanical gripper should be connected;

wherein the mechanical gripper is configured to boot-up and to start communicating with the GUI element thereby enabling detection of the of the mechanical gripper by the third-party software;

wherein the display device is configured to display a configuration interface enabling configuration of one or more physical parameters relating to the mechanical gripper, including a maximum closing force of the mechanical gripper; and

wherein the controller is configured to receive, via the program tab, programming for a task to be performed by the robot using the mechanical gripper.

15. The robot of claim 9, wherein the third-party hardware comprises a vision system;

wherein the controller is configured to receive data representing selection of the GUI element in the installation tab;

wherein the display device is configured to display information about how the vision system should be connected to the robot, information about how coordinates of the vision system relate to coordinates of the robot, and information about one or more parts to identify based on the vision system, the program tab comprising a GUI element enabling programming the robot to move to an identified part; and

wherein the controller is configured to generate a robot program for performing a work cycle by moving the identified part to a predefined position, the robot program for performing a mathematical transformation between the coordinates of the vision system and the coordinates of the robot for the identified part.

16. The robot of claim 9, wherein the third-party hardware comprises an interface to industrial equipment;

wherein the display device is configured to display information in the installation tab about connecting the robot to the industrial equipment; and

wherein the controller is configured to incorporate functions into the programming tab to include triggering events or functions in the industrial equipment.