CANopen, a key factor in motor control systems for seeding applications

José Antonio Pulido, DOGA Spain

A motor control system using the CANopen CiA-402 device profile was developed for controlling and monitoring individually the seed dispensing for a planter row unit. The paper describes relevant aspects in the integration of the CANopen protocol in electronic motors for planter systems. We discuss the flexibility the protocol provides for covering specific applications like seeding, the advantages of using a standardized driver profile like CiA-402 instead of manufacturer-specific designs, the mode of operations: speed and position modes for closed-loop regulation, the network configuration, feedback signals for monitoring the motor performance and the emergency mechanism established between components. The paper is mainly focused on describing the most relevant decisions in terms of design that were made during the integration of CANopen communications in an electronic motor for seeding applications.

Overview

A traditionally planting system provides a complete control for planting using seed tube delivery. Nowadays, the advancements in control systems improve significantly the accurate plant spacing for obtaining the desired population. A seeding machine is also equipped with other components like fill disk, seed sensor and in particular the individual row control in charge of controlling and monitoring individually electronic motors for seed dispensing.

As an electric and electronic motor manufacturer, we successfully achieved the integration of an electronic solution on the client planter system. The solution was brushed DC motors with a digital Hall sensor for a closed-loop speed regulation and position control using a CAN communication interface. In this case CANopen was the most suitable protocol for the application, and an innovation for our commercial products mainly based on the CAN J1939 protocol, LIN interface, etc. The high-level protocol should be compliant to the standard CiA-301 [1], parts of the drive profile CiA-402 [3] for motion control and manufacturer-specific motor profile.

In the next sections the different technical aspects faced during the integration will be discussed.

Network management

Each individual row control is designed to command a maximum number of two motors as slave nodes in the network. In this case a configurable node-ID would be desirable in order to guarantee the system flexibility. The CANopen network management (NMT) doesn't support dynamic reassignment identification as done in CAN J1939 protocol.

Regarding the network initialization, the standard only proposes one example; open to specific implementations. That uncovered functionality leads to face first important integration decisions. In the example the master node, once in normal operation, resets and starts each slave node. Conversely the solution is highly dependent on the master capability and preferred mechanism.

For the current development, it was decided that each slave node will transmit periodically a boot-up message after the initialization (see Figure 1). Once the master node detects them, it will proceed with the object configuration, and the motors will

stop sending the boot-up message after the reception of a valid message.

A detailed initialization specification would have made the integration phase easier.

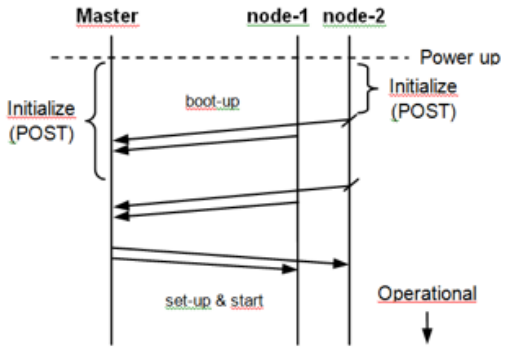

Figure 1: Initialization of two nodes via boot-up messages

Communication requirements

The motor control requires at least the standard SDO and network management services. Beside that the emergency service could provide a valuable fault protect mechanism that it will be discussed later on. The efficiency and real-time transfers that provides the PDO service is not absolutely necessary for the seeder applications. Thus the motor control and feedback signals will be accessed by only SDOs. That could affect the system scalability in case of a growing amount of nodes in the CAN bus to be controlled. In any case, the current network configuration based on two nodes for each control module can manage the traffic flow with a normal level of bus load.

Memory resources

The ROM memory usage in a CANopen stack is mainly allocated to cover the read/write access to variables in the object dictionary for all data types and the communication protocols like SDO and PDO services. The SDO impact on the ROM memory is practically similar to other protocols, e.g. the CAN J1939 transport protocol layer (BAM, RTS/CTS messages).

On the other hand, mandatory objects with write access lead to generate them on RAM memory. Compared to other stacks, these objects would not be needed in the application although they do not occupy a considerable amount of memory. If the device supports calibration capability, the number of manufacturer objects should be taken into account in the estimation of RAM resources.

As a result, the CANopen stack integration does not require more memory usage than other solutions. The memory size could be affected when additional features such as a flash bootloader and/or calibration are added.

Memory access authentication

The CANopen stack does not provide an authentication mechanism for memory access like CAN UDS or J1939-73 diagnostic application layers. Consequently, manufacturer objects have to be created to accomplish a password or seed/key procedure. However, the standard CiA-301 defines a simple technique for storing/restoring device parameters using a pre-defined signature/password. That known signature does not prevent unauthorized access to the device memory after all. In terms of security access the device manufacturer should be responsible for the password number definition.

Manufacturer vs standard objects

A device conforming to the standard has to implement mandatory objects even if the application defines alternative manufacturer control objects for convenience. As an example, the standard CiA-402 defines a generic motor control (PDS FSA) that could be highly simplified (see Figure 2). Custom designs based on just the mode of operation, the velocity/position target and an error management would be desired on the master side. That simplified version provides required results, but contradictory to the standard procedure.

On the other hand a particular implementation not following exactly the standard has an impact on the systems' flexibility because it makes the access to new motor suppliers difficult.

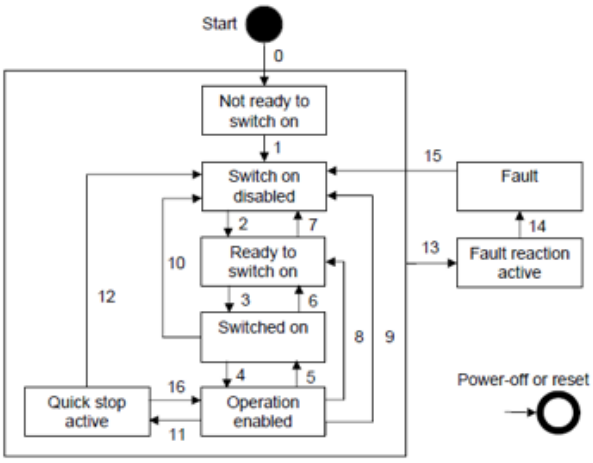

Figure 2: Power drive system

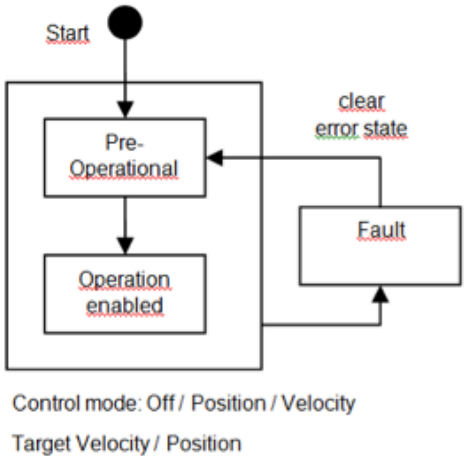

Figure 3: Simplified motor control system

Motor drive system

Depending on the master network capabilities, the previous control description could be adapted to not support all motor functions. Indeed the motor operation could be enabled via automatic local transitions without any master command on the network.

At this point, the control will take actions only according to the selected operation mode and set-points, so that the provided objects (Controlword/Statusword) for controlling the motor turn into very generic objects.

As an example, Figure 3 depicts a simplified control diagram where there are only three defined states. After the motor powers on, it will perform automatic communication and driver initialization.

Once the start-up sequence is finished, the state machine makes a transition from 'Pre-Operational' to 'Operation Enabled' state, ready to rotate the shaft. If the operation mode is "Off", the controller has to disable the high-level power, so the motor is free to rotate. Otherwise the controller has to enable it and process target inputs. In case of a failure condition, there shall be a transition to 'Fault' state where the high-level power shall be disabled to prevent any potential damage. In this scenario a manufacturer object for clearing motor error state shall be needed for a complete definition of motor control.

The example on Figure 3 makes it easier for the master side to control it, but it's not completely compliant to the standard CiA-402.

Internal warning and error

The control logic shall detect, report and recover not only internal failures such as load level error, stall error, sensor error…, but also internal warnings to prevent any potential damage during operation. The CANopen emergency service provides a mechanism to manage error events; however, if the master does not support the emergency message consumer, a polling process for requesting error status shall be defined between master/slave. Moreover the error object should not contain warning conditions that do not lead to motor failure, so if needed, a specific manufacturer object should be defined to report them.

The controller should protect the motor against damage mostly caused by overload, stall condition, short-circuit, under-voltage, over-temperature and sensor error. The overload protection controls it from exceeding the maximum nominal current along the time and it prevents the motor from exceeding its temperature rating. The stall condition detects when the motor current is higher than the maximum current limit when for example; some rocks interfere with the seeder. On all failure cases, the controller shall shut down the output stage and report it in the error status object. If the controller receives a command for clearing the error state and the condition is resolved, the controller should recover the nominal operation.

Emergency stop

The emergency stop signal for seeding machines is a safety feature that automatically stops the electronic motor system preventing any potential hazard/ failure. Normally, the emergency stop is implemented by a dedicated physical line. In case of no signal availability, the emergency stop could be implemented using the CANopen heartbeat service.

In this scenario, electronic motors as slave nodes are continuously monitoring the reception of the heartbeat message transmitted by the master node. If the next heartbeat message is not received before the configured consumer time, the controller will react immediately disabling the motor operation and reporting the heartbeat error. After a clearing command, the controller will be ready to receive the first and consecutive heartbeat messages for nominal operation.

Speed and position controllers

The standard CiA-402 defines a collection of profiles that perform particular control functions. That means that an ECU may serve as one or more controllers without the need to associate a specific identification on the network. The seeding application requires at least two controller profiles: velocity mode and relative position mode.

In velocity mode the controller shall regulate the motor speed at the required velocity using a PID algorithm. The application requires a high accuracy in the range from 20 RPM to 30 RPM for nominal torque in order to assure a high seed spacing accuracy rotating a flat-type seed disk. In this case, the proposed solution was an electronic brushed DC motor (see Figure 4).

Figure 4: Electronic brushed DC motor

Moreover the planter system is capable of controlling individually up to 48 motors with the objective of maintaining target population especially on the curve. For this purpose the machine has two radars, one on the left and one on the right side to receive the speed of the row unit and perform the curve compensation. The compensation consists in modifying the motor velocity based on the ground speed of that respective row.

On the other hand, if using the relative position mode, the seeding machine can rotate the seed disk to the desired position in test modes like fill disk operation and multiple revolution tests. The fill disk operation consists in turning the disk one complete revolution with a maximum velocity limit during the change in position. Both, the target and the actual positions are expressed in armature units. That means a disk revolution with 81:1 gear rating will be done setting 81 revolutions to the target position.

Both modes require a closed-loop PID controller to regulate the motor (velocity/ position). The motor control has to eliminate steady state errors on step references changes with a damped oscillator response. For that purpose the controller implements a common PID algorithm. In the case of velocity mode the error is the difference between the velocity demand and the actual motor velocity. The velocity demand is the velocity target with predefined acceleration deceleration linear ramps and the actual velocity is the average motor velocity. The velocity acceleration/deceleration parameters are defined in the standard CiA-402 as the maximum rate of change over time when increasing/decreasing speed. For this application the acceleration/deceleration rate has to be 60 RPM / sec.

The PID controller also implements an integral anti-windup circuit that switches off the integral action when the control effort is over the maximum saturation level. Moreover the range of target velocity is restricted inside minimum and maximum valid limits to avoid unintended values. In this case the motor always turn clockwise (CW) if viewed from the shaft end and the speed range goes from 3 RPM to 60 RPM.

As an example, Figure 5 shows the actual velocity following the target along different set-points changes. When reaching the target velocity with certain tolerance, the controller shall notify the event activating the corresponding bit of the Statusword object.

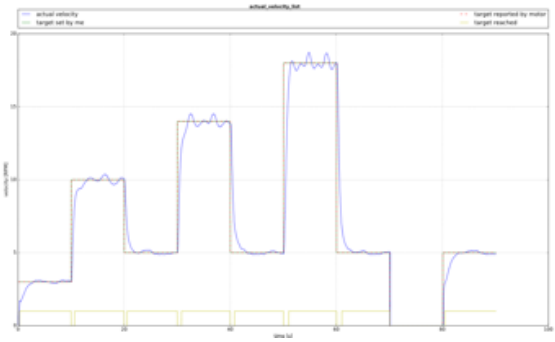

Figure 5: Actual vs target velocity in different set-point changes

In addition to the common feedback signals like actual velocity and position, the electronic motor will provide extra physical signals such as battery voltage, motor current and internal motor temperature in order to monitor its performance during operation.

In this case, the SDO service is used to poll individually each feedback signal. By definition, polling mechanism does not assure data correlation because multiple values are measured sequentially and it produces unnecessarily high bus load. However, those limits do not impact in the processing of the current application because synchronous measurements are not required for the process flow.

Calibration data

As described early, each individual row control has two motors connected on the same CAN bus. The parameter “node-ID” should be the unique one and shall be changed for each ECU’s motor software. The current development implements an EEPROM emulation using the flash memory where multiple flash sectors are used in alternation to record parameter changes.

In general, the parameter “node-ID” could be modified directly in the flash memory of the hex file or by the online calibration.

The online calibration provides the write/read memory access of particular parameters in the object dictionary at runtime. Once desired device parameters are determined individually, they can be stored in the ECU’s flash memory or restored to default values using store/restore objects defined in the standard CiA-301. In addition of node-ID parameter, calibration data also could be baud rate configuration, PID constants, maximum/minimum protection limits, acceleration/deceleration ramps, etc.

Apart from the re-flashing capability for calibration data or for application software download via CANopen bootloader [2], it will be desirable to allow an EOL (End-of-line) tester to execute specific tests for the verification of ECU’s hardware modules before the electronics leave the supplier phase. Manufacturer-specific objects should be created to provide a custom service with an authentication level to execute tests/ routines similar to the CAN UDS Remote Routine Control.

Conclusion

The main aspects faced during the integration of CANopen communications in a commercial product for seeding applications has been shortly reviewed.

The standard CiA-402 specifies several device profiles, such as, velocity and position modes that meet perfectly the application control needs, although a few manufacturer specific decisions have been taken along the project for achieving a final product (see Figure 6).

Figure 6: Seeder row units

References

[1] CiA DS 301, CANopen application layer and communication profile
[2] CiA DS 302, CANopen configuration and program download
[3] CiA DSP 402, CANopen drive and motion control profile

José Antonio Pulido
DOGA S.A.
Autovía A-2 Km. 583
08630 Abrera, Barcelona, Spain
Tel.: +34-93-770-46-00
Fax: +34-93-770-44-80
josea.pulido@doga.es
www.doga.es