

UNIVERSITATEA TEHNICĂ „Gheorghe Asachi” din IAȘI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DOMENIUL: Calculatoare și Tehnologia Informației
SPECIALIZAREA: Calculatoare Încorporate

SISTEM DE MONITORIZARE ȘI CONTROL MULTI-NOD ANALIZOR CAN

Student:
Cristian GORCEAC
Ionut-Sebastian LUCHIAN
Rafael-Cristian TUDUSCIUSC

Iași, 2026

Cuprins

1	Introducere	2
1.1	Descriere	2
1.1.1	Arhitectura Sistemului și Distribuția Echipelor	2
1.1.2	Schema Logică a Transmisiei	2
1.2	Placa de dezvoltare	2
1.3	Arhitectură Microcontroler (MCU)	3
1.4	Caracteristici Hardware și Periferice	3
1.5	Ecosistem de Dezvoltare	3
1.6	Implementare Tehnică	3
2	Implementarea Analizorului CAN-FD pentru Monitorizarea Magistralei	5
2.1	Rolul Analizorului CAN în Arhitectura Sistemului	5
2.2	Arhitectura Generală a Sistemului CAN-FD	5
2.2.1	Descrierea Nodurilor din Rețea	5
2.2.2	Diagramă Arhitecturală a Rețelei CAN-FD	5
2.3	Structura Software a Analizorului CAN-FD	6
2.3.1	Organizarea Aplicației	6
2.3.2	Configurarea Sigură a Transceiverului CAN	6
2.4	Configurarea FlexCAN-FD	6
2.4.1	Parametri de Comunicație	6
2.5	Recepția și Interpretarea Cadrelor CAN	6
2.5.1	Structura Logică a Cadrului Monitorizat	6
2.5.2	Decodarea Mesajelor	6
2.6	Gestionarea Erorilor	7
3	Concluzii și Direcții de Dezvoltare	8
3.1	Concluzii	8
3.2	Direcții de Dezvoltare Viitoare	8
3.2.1	Optimizarea Consumului de Energie	8
3.2.2	Implementarea unei Interfețe Om-Mașină (HMI)	9
3.2.3	Extinderea Rețelei CAN	9
3.2.4	Algoritmi de Control Avansati	9

Capitolul 1. Introducere

1.1. Descriere

Acest proiect implementează un sistem de control distribuit utilizând protocolul de comunicație industrială **CAN (Controller Area Network)**. Obiectivul principal este transferul de date de la nodurile de intrare (senzori) către nodurile de execuție (actuatori), sub supravegherea unui nod dedicat de monitorizare. Întregul sistem este construit pe platforma **NXP FRDM-MCXN947**.

1.1.1. Arhitectura Sistemului și Distribuția Echipelor

Sistemul este compus din 5 noduri, fiecare fiind gestionat de o echipă specifică:

1. **Potențiometru (Input):** achiziția analogică (ADC) și transmiterea poziției către rețea pentru controlul unghiular sau de intensitate.
2. **Senzor de Temperatura/Umiditate (Input):** monitorizarea condițiilor de mediu și partajarea acestor parametri pe magistrală.
3. **Monitor:** interceptarea pachetelor. Acesta nu modifică datele, ci analizează traficul pentru validarea comunicării între celelalte noduri.
4. **Servomotor (Output):** Interpretează datele primite (de la potențiometru și senzor) și generează semnalul PWM necesar poziționării mecanice.
5. **LED RGB (Output):** Oferă feedback vizual pe baza datelor de la senzori.

1.1.2. Schema Logică a Transmisiei

Figura 1.1 ilustrează modul în care datele circulă de la cele două surse de intrare către restul componentelor sistemului.

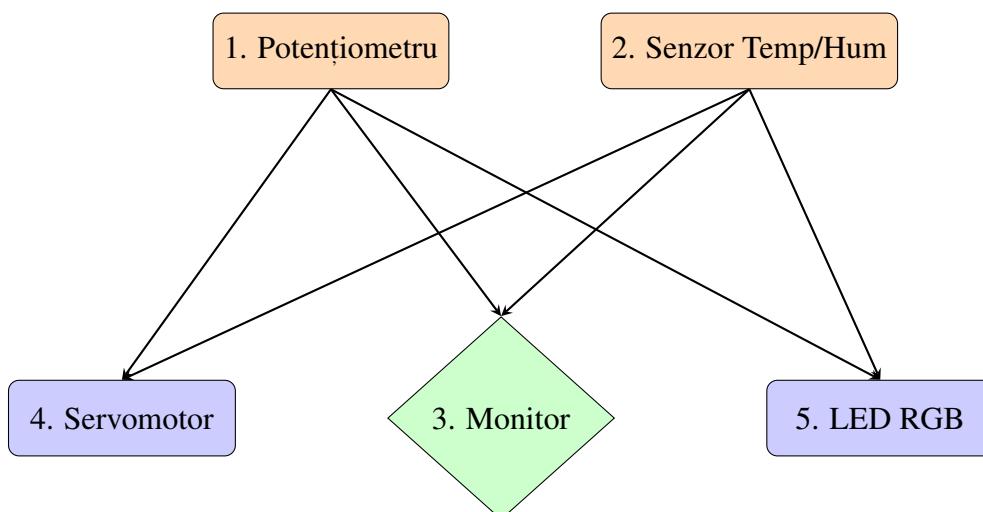


Figura 1.1. Schema de distribuție

1.2. Placa de dezvoltare

Placa de dezvoltare **FRDM-MCXN947** este o platformă compactă și scalabilă, concepută pentru evaluarea microcontrolerelor din seria **MCX N94x**. Aceasta este ideală pentru aplicații de tip Edge Computing, automatizări industriale și Internet of Things (IoT).

1.3. Arhitectură Microcontroler (MCU)

Nucleul plăcii este reprezentat de MCU-ul **MCX N947**, care dispune de următoarele caracteristici:

- **Dual-Core:** 2x Arm® Cortex®-M33 rulând la frecvențe de până la 150 MHz.
- **Accelerator AI/ML:** Include unitatea de procesare neurală **eIQ® Neutron NPU** pentru sarcini de inteligență artificială la nivel hardware.
- **Securitate:** Subsistem de securitate **EdgeLock® Secure Enclave** (Core Profile).
- **Memorie:** Până la 2 MB de memorie Flash și 512 KB de memorie SRAM cu paritate/ECC.

1.4. Caracteristici Hardware și Periferice

Placa FRDM-MCXN947 integrează o gamă largă de componente pentru prototipare rapidă:

- **Conecțivitate Ethernet:** Port RJ45 integrat pentru aplicații de rețea.
- **Interfețe USB:** Port USB Type-C High-Speed pentru date și alimentare.
- **Expansiune:** Compatibilitate cu shield-uri **Arduino Uno R3** și conectori **MikroBus™** pentru senzori și actuatori.
- **Debug Integrat:** On-board **MCU-Link** debugger bazat pe CMSIS-DAP, eliminând necesitatea unui programator extern.
- **Interfață Audio:** Suport pentru ieșiri audio de tip MQS (Medium Quality Sound).

1.5. Ecosistem de Dezvoltare

Suportul software este asigurat prin **MCUXpresso Developer Experience**, oferind:

1. **IDE:** MCUXpresso IDE, VS Code (cu extensii NXP) sau IAR/Keil.
2. **SDK:** Drivere periferice optimizate, stive de protocoale și exemple de cod (RTC, PWM, Ethernet).
3. **Configurare:** Instrumente grafice pentru rutarea pinilor și configurarea ceasurilor (Pins, Clocks, Peripherals Tool).

1.6. Implementare Tehnică

Pentru realizarea comunicării, s-au utilizat următoarele resurse hardware ale microcontrolerului MCX N947:

- **FlexCAN:** Configurarea baud-rate-ului și a filtrelor de acceptare (Hardware Acceptance Filters).
- **GPIO & PWM:** Pentru controlul direct al servomotorului și al LED-ului.
- **ADC:** Pentru conversia semnalului analogic de la potențiometru.

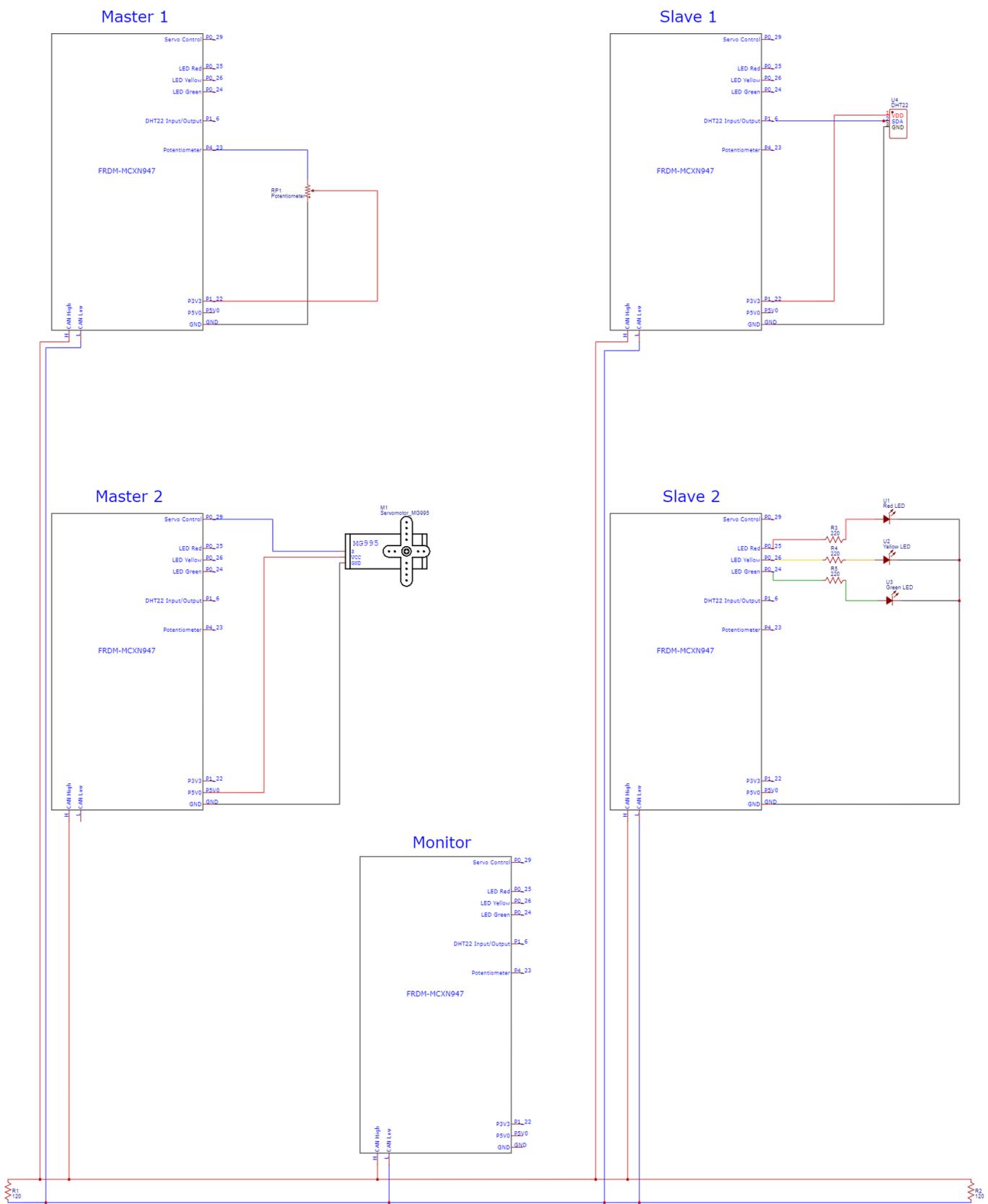


Figura 1.2. Schema Arhitecturii Sistemului

Capitolul 2. Implementarea Analizorului CAN-FD pentru Monitorizarea Magistralei

2.1. Rolul Analizorului CAN în Arhitectura Sistemului

Analizorul CAN-FD reprezintă un nod pasiv al rețelei CAN, având rolul de a monitoriza și interpreta traficul de date dintre nodurile active ale sistemului, fără a influența procesul de arbitrage sau fluxul comunicației.

Acest nod este utilizat exclusiv pentru:

- validarea corectitudinii mesajelor CAN-FD;
- depanarea și analiza comunicației distribuite;
- verificarea rutării logice sursă–destinație;
- monitorizarea în timp real a datelor transmise.

2.2. Arhitectura Generală a Sistemului CAN-FD

2.2.1. Descrierea Nodurilor din Rețea

Rețeaua CAN-FD implementată este formată din următoarele noduri:

- **Nod Master – Potențiometru**: generează comenzi procentuale;
- **Nod Master – DHT22**: transmite date de temperatură și umiditate;
- **Nod Slave – Servo Motor**: acționează mecanic;
- **Nod Slave – LED**: semnalizare vizuală;
- **Analizor CAN-FD**: monitorizează întreg traficul.

2.2.2. Diagramă Arhitecturală a Rețelei CAN-FD

Figura 2.1 prezintă structura logică a sistemului și poziționarea analizorului CAN.

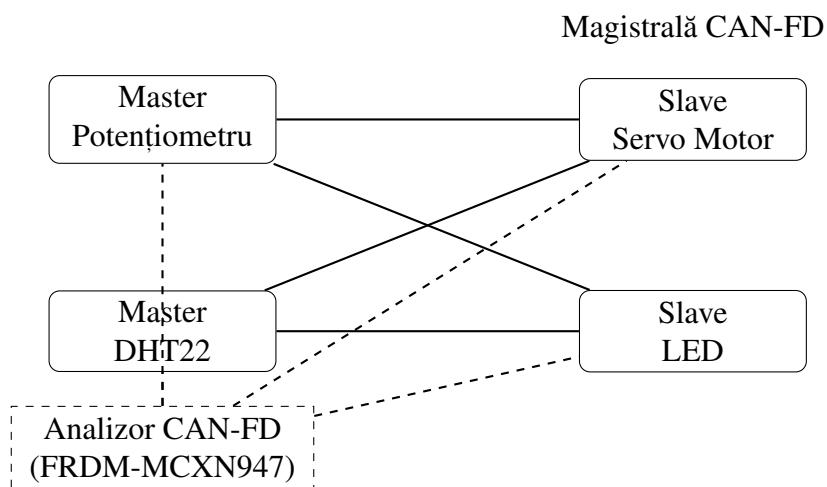


Figura 2.1. Diagramă arhitecturală a sistemului CAN-FD cu analizor

2.3. Structura Software a Analizorului CAN-FD

2.3.1. Organizarea Aplicației

Aplicația este structurată modular și cuprinde următoarele etape:

- inițializarea hardware a plăcii FRDM-MCXN947;
- configurarea perifericului FlexCAN-FD;
- configurarea sigură a transceiverului CAN (PHY);
- activarea modului promiscuu;
- recepția și decodarea mesajelor CAN-FD.

2.3.2. Configurarea Sigură a Transceiverului CAN

Funcția FLEXCAN_PHY_Config_Safe este utilizată pentru activarea controlată a transceiverului CAN. Aceasta trimite un mesaj de test pe magistrală și așteaptă confirmarea ACK din partea celorlalte noduri.

Pentru a evita blocarea sistemului, este utilizat un mecanism de timeout. În lipsa confirmării, transmisia este anulată, iar nodul continuă funcționarea în regim de monitorizare.

2.4. Configurarea FlexCAN-FD

2.4.1. Parametri de Comunicație

Perifericul FlexCAN este configurat cu următoarele valori:

- rată nominală (arbitrare): 500 kbps;
- rată de date CAN-FD: 2 Mbps;
- format standard al cadrului CAN;
- activarea *Individual Masking*.

Masca individuală este setată la valoarea 0x000, permitând receptia tuturor identificatorilor CAN, transformând nodul într-un analizor complet.

2.5. Recepția și Interpretarea Cadrelor CAN

2.5.1. Structura Logică a Cadrului Monitorizat

Mesajele CAN sunt interpretate conform următoarei structuri:

- **ID CAN**: destinația mesajului;
- **Byte 0**: identificatorul sursei;
- **Byte 1**: valoarea transmisă.

Această structură permite separarea clară a rolurilor în rețea.

2.5.2. Decodarea Mesajelor

În funcție de combinația sursă–destinație, analizorul afișează mesaje descriptive pentru:

- poziția potențiometrului;
- temperatura și umiditatea de la DHT22;
- comenzi destinate servo motorului sau LED-urilor.

2.6. Gestionarea Erorilor

Erorile de recepție sunt raportate prin interfața de debug serial, iar sistemul revine automat în starea de ascultare, fără întreruperea funcționării.

Capitolul 3. Concluzii și Direcții de Dezvoltare

3.1. Concluzii

Proiectul prezentat a demonstrat cu succes realizarea unui sistem embedded distribuit pentru monitorizare și control, bazat pe microcontrolerul **FRDM-MCXN947** și pe protocolul de comunicație industrial **CAN / CAN-FD**. Arhitectura propusă a permis interconectarea mai multor noduri funcționale (senzori, actuatori și analizor CAN) într-o rețea robustă, deterministă și ușor extensibilă.

Un element central al proiectului îl constituie integrarea nodului de achiziție analogică pentru potențiometru, care a evidențiat capabilitățile avansate ale perifericului **LPADC**. Utilizarea rezoluției de **16 biți**, împreună cu mecanismele de auto-calibrare hardware, a condus la obținerea unor valori stabile și precise, adecvate pentru aplicații de control în timp real.

De asemenea, implementarea algoritmilor software de filtrare și cuantizare a valorilor analogice a avut un impact direct asupra stabilității comunicației CAN. Prin limitarea variațiilor mici ale semnalului, s-a redus semnificativ numărul de transmisii redundante, prevenind încărcarea inutilă a magistralei și asigurând un comportament predictibil al actuatorilor.

Principalele realizări ale proiectului sunt sintetizate mai jos:

- **Precizia achiziției analogice:** configurarea LPADC cu rezoluție maximă și calibrare automată a permis eliminarea erorilor de offset și drift, oferind date fiabile pentru control.
- **Stabilitatea comunicației CAN:** utilizarea filtrării de tip „snapping” la pași de 5% a redus jitter-ul digital, contribuind la mișcări fluide ale servomotorului și la un control coerent al LED-ului RGB.
- **Robustetea sistemului distribuit:** protocolul CAN-FD a asigurat transmiterea corectă a mesajelor între cele cinci noduri ale sistemului, inclusiv în prezența unui nod pasiv de tip analizor CAN.
- **Modularitatea arhitecturii:** separarea clară între noduri de tip master, slave și monitor a permis testarea și validarea fiecărei componente în mod independent.

În concluzie, ecosistemul NXP MCX s-a dovedit adecvat pentru dezvoltarea rapidă a unor aplicații embedded distribuite, oferind un echilibru optim între consum energetic, performanță și flexibilitate software. Soluția implementată poate fi considerată o bază solidă pentru aplicații de automatizare și control industrial.

3.2. Direcții de Dezvoltare Viitoare

Deși sistemul realizat îndeplinește obiectivele propuse, acesta oferă multiple posibilități de extindere și optimizare, atât din punct de vedere hardware, cât și software.

3.2.1. Optimizarea Consumului de Energie

O direcție importantă de dezvoltare o reprezintă reducerea consumului energetic prin utilizarea modurilor avansate de economisire a energiei disponibile pe microcontrolerul MCXN947. Nodul de achiziție analogică ar putea utiliza mecanisme de tip *Wake-on-ADC*, activând procesorul doar atunci când variația tensiunii depășește un prag prestabilit, ceea ce ar conduce la un consum mediu semnificativ mai redus.

3.2.2. Implementarea unei Interfețe Om–Mașină (HMI)

Integrarea unei interfețe grafice, bazate pe un afișaj TFT sau OLED, la nivelul nodului de monitorizare, ar permite afișarea în timp real a parametrilor sistemului. Vizualizarea grafică a poziției potențiometrului sau a valorilor de temperatură și umiditate ar îmbunătăți considerabil experiența utilizatorului și ar facilita procesul de depanare.

3.2.3. Extinderea Rețelei CAN

Arhitectura actuală poate fi extinsă prin adăugarea unor noduri suplimentare, precum senzori de proximitate, senzori de presiune sau module de comunicație wireless. Integrarea unui gateway CAN–Wi-Fi sau CAN–Bluetooth ar permite conectarea sistemului la o platformă **IoT Cloud**, facilitând monitorizarea și controlul de la distanță.

3.2.4. Algoritmi de Control Avansați

Pentru nodul de acționare al servomotorului, o îmbunătățire semnificativă ar consta în implementarea unui regulator de tip **PID** (Proporțional–Integral–Derivativ). Acest tip de control ar permite o poziționare mult mai precisă, reducerea timpului de răspuns și eliminarea oscilațiilor apărute la variații brusăte ale valorilor de referință.

Bibliografie

- [1] NXP Semiconductors, *FRDM-MCXN947 Board User Guide*, Document Number: FRD-MMCXN947UG.
- [2] NXP Semiconductors, *MCX N Series Microcontrollers Reference Manual*, rev. 1, 2024.
- [3] NXP Semiconductors, *MCUXpresso SDK API Reference Manual*, capitolele *LPADC Driver* și *FlexCAN Driver*.
- [4] ISO, *ISO 11898-1:2015 — Controller Area Network (CAN)*, standard pentru protocolul CAN și CAN-FD.
- [5] NXP Semiconductors, *Application Note: Configuring LPADC for High Precision Measurements on MCX Series*.